ADDRESSING STIFFNESS-INDUCED CHALLENGES IN MODELING AND IDENTIFICATION FOR RIGID-BODY SYSTEMS WITH FRICTION AND IMPACTS

Mathew Halm

A DISSERTATION

in

Mechanical Engineering and Applied Mechanics

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2023

Supervisor of Dissertation

Michael Posa, Assistant Professor of Mechanical Engineering and Applied Mechanics

Graduate Group Chairperson

Prashant K. Purohit, Professor of Mechanical Engineering and Applied Mechanics

Dissertation Committee

Vijay Kumar, Nemirovsky Family Dean of the School of Engineering and Applied Science and Professor of Mechanical Engineering and Applied Mechanics
Michael Posa, Assistant Professor of Mechanical Engineering and Applied Mechanics
Dinesh Jayaraman, Assistant Professor of Computer and Information Science

ADDRESSING STIFFNESS-INDUCED CHALLENGES

IN MODELING AND IDENTIFICATION

FOR RIGID-BODY SYSTEMS WITH FRICTION AND IMPACTS

ACKNOWLEDGEMENT

Completing a Ph.D. comes at the conclusion of a decades-long educational journey with many challenges, complications, and surprises, along which I have received immense support and contribution from countless communities and individuals.

I have been extremely lucky to be advised by Prof. Michael Posa for the full duration of my PhD. As his first doctoral student, I have received innumerable hours of technical and professional guidance from him, especially in our early years at Penn before the DAIR Lab rapidly expanded into the community it is today. Michael overcame my stubborn, instinctual resistance to mentorship with the towering intellect, compelling persuasion, and unending kindness that draws students even from outside the lab to seek his advice.

I have also been strongly influenced by many others in Penn's faculty. Transitioning my background in mechanics and mathematics to applications in optimization and machine learning was aided by the perspective of and collaboration with Prof. Nikolai Matni. I am grateful for Dean Vijay Kumar's and Prof. Dinesh Jayaraman's time and effort in evaluating and providing feedback on this work, both informally and especially more recently in their official capacities as Chair and Reader on my dissertation committee.

This dissertation was conducted with earnest, close, and fruitful collaboration with other members of the DAIR Lab. My first collaborators, Sam Pfrommer and Mihir Parmar, made critical and foundational contributions beyond their years during their bachelor's and master's programs, respectively. They have since unsurprisingly earned great success in their further careers in academia and industry. Much of my later work was conducted closely with and has benefitted from my fellow PhD student Bibit Bianchini and her incredible work ethic and sharp analysis. I am also grateful for my collaboration with and the ongoing work of now-Professor Wanxin Jin and fellow PhD student Alp Aydinoglu on connecting ideas in this thesis to real-world control problems. Beyond my co-authors, the remainder of the DAIR Lab has been a valuable source of camaraderie, technical influence, and valuable feedback.

I greatly owe my acceptance into Penn to the help and influence of my undergraduate research advisor, Prof. Seth Hutchinson, through whom I discovered a passion for research and first met Michael.

Before working with Seth, I was a computer science student feeling apathetic about my interest in software engineering. I found passion in mechanics, control theory, and robotics through many dear and influential friends in Student Space Systems at UIUC.

Long, long before my engineering education started, I was very privileged to grow up in a house full of scientists and engineers without which I would have never found my calling. Fond and formative memories include learning JavaScript from my sister in middle school; learning chemistry techniques from my dad for my 8th grade science fair project; and my mom's relentless search for every accelerated educational opportunity she could find for me throughout my childhood.

Finally, I am thankful for the many close friends made in the Penn community, for their years of support and the precious memories of time well-spent together, especially from my dear and loving partner, Laura.

ABSTRACT

ADDRESSING STIFFNESS-INDUCED CHALLENGES
IN MODELING AND IDENTIFICATION
FOR RIGID-BODY SYSTEMS WITH FRICTION AND IMPACTS

Mathew Halm

Michael Posa

Imperfect, useful dynamical models have enabled significant progress in planning and controlling robotic locomotion and manipulation. Traditionally, these models have been physics-based, with accuracy relying upon manual calibration only feasible in laboratory environments. As robotics expands into complex real-world applications, models of unknown environments must instead be automatically fit to limited data. One major challenge is modeling frictional contact, especially during collisions involved in common robotics tasks. Rapid deformation under impact manifests as extreme sensitivity to initial conditions and material properties. Thus, even slight errors in state estimation and system identification can lead to significant prediction errors. Consequently, model inaccuracy or the *sim-to-real gap* often hinders the development of performant robotics algorithms.

Physical models can be optimized using advanced techniques to overcome these challenges, but such methods have limited tractability when a large number of parameters are unknown. Furthermore, even given accurate parameters, roboticists often make inaccurate rigid-body approximations to reduce the computational burdens of physical simulation to meet faster-than-real-time requirements. An alternative *black-box* approach, in which models are learned from scratch, has attempted to address these issues for instance using deep neural networks (DNN's). While DNNs in theory can capture any dynamical behavior, they empirically struggle with the stiff behaviors associated with contact.

This dissertation instead focuses on scaling physical model identification to the high-dimensional setting and quantifying the limited accuracy of low-fidelity models. We consider rigid bodies under-

going rigid contact, for which infinite stiffness is represented as constrained optimization. By careful treatment of these constraints, we demonstrate that infinitely-stiff dynamics can be identified by optimizing a non-stiff objective. In conjunction, we use DNN's in a *white-box* setting to model physical quantities, specifically reconstructing geometries from scratch. We then consider how rigid-body collision models lack the fidelity to correctly predict outcomes of nearly-simultaneous impacts—such as heel and toe strikes during a footstep. We develop a theoretical basis to capture partial knowledge of impact events as uncertain set-valued outcomes, and again use numerical optimization to compute approximations of such sets.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER 1

Introduction

Imperfect but useful dynamical models have long enabled improvements in planning and control of robotic locomotion and manipulation. Traditionally, roboticists have relied on high-accuracy, physics-based models of tightly-controlled laboratory environments. In these settings, obtaining such a model is often a manual process, in which physical properties such as masses, lengths, and shapes are often preliminarily calculated from design documents, and a handful of important quantities may be further calibrated with abundant measurements from precise instrumentation. However, as robotics shifts toward complex real-world applications in unknown environments, model accuracy requirements are increasing; calibration data is becoming scarcer; and the model-fitting process must be fully automated. In response, recent work in the field has focused on how empirical, data-driven models can be automatically fit to the limited available data. In spite of these efforts, model inaccuracy or the *sim-to-real gap* remains a common bottleneck in both machine learning-based (Ibarz et al., 2021) and mechanics-based robotics algorithms (Wensing et al., 2022).

Frictional contact, the physical process driving these robotic tasks, is a principal source of these challenges. Impacts among robots and their surroundings are particularly difficult to model. When objects collide, materials deform on an imperceptibly small spatial and temporal scale, preventing interpenetration. The underlying material property driving this rapidity, *mechanical stiffness*, causes multiple forms of *numerical stiffness* in the equations of motion of these systems (Stewart, 2000). Even small changes in initial conditions and material properties generate large changes in real-world outcomes. Additionally, measurements of the velocities are extremely sensitive to the time that they are recorded, as they change near-instantaneously during impact (Fazeli et al., 2017b). Accordingly, small errors in state estimation and parameter identification produce large prediction error, even over a small time horizon (Chatterjee, 1997; Ajay et al., 2018; Ibarz et al., 2021). These properties become significant problems when fitting a model of a real system from noisy sensor measurements.

In settings where only a handful of parameters are unknown, the field has derived several tools

to overcome these challenges with purely physics-based models. When many more datapoints than unknowns are available (known as *underparameterization*), statistical analysis can guarantee that the parameters that minimize objectives akin to nonlinear least-squares are highly accurate (Belkin et al., 2019; Bianchini et al., 2022). While the stiffness in the dynamics makes the process of finding such parameters numerically challenging, the low-dimensional setting allows such difficulties to be overcome by applying advanced optimization techniques, such as second-order (Kolev and Todorov, 2015), constrained (Fazeli et al., 2017b), or global (Heiden et al., 2020) methods. Utilizing these techniques can be difficult, as their setup require extensive human knowledge of the robot and its surroundings, and fitting model parameters requires expertise in both mechanics and optimization. Prior art has not explored how higher-dimensional unknowns, such as reconstructing geometries from scratch, could be effectively embedded into these techniques. Furthermore, the outcomes of collisions have a complex relationship with the material properties and deformation dynamics of the involved objects (Chatterjee, 1999). These behaviors are in general are not tractable to simulate under the faster-than-real-time computational requirements of real-world robotics problems. Under these constraints, roboticists typically only have access to rigid-body models, a tractable but coarse approximation of contact mechanics in which objects do not deform. Therefore, even excellent parameter identification algorithms produce models that are fundamentally limited by these inaccuracies (Fazeli et al., 2017a).

Due in part to these difficulties, an alternative approach has emerged, in which the entirety of the dynamical behavior of robots and their environments are learned from scratch (Ajay et al., 2019; Janner et al., 2019; Chua et al., 2018; Nagabandi et al., 2019; Ajay et al., 2018; Ibarz et al., 2021). In this setting, often referred to as *black-box* modeling, the dynamics are modeled as a universal function approximator such as a deep neural network (DNN); such models have a vast number of unknown parameters which give them the flexibility to capture any possible dynamical behavior. However, particularly in the case where the dynamics to be learned are stuff, this flexibility comes as significant cost. As these tools are often used in an *overparameterized* regime where the parameters outnumber available data, many distinct models may exactly fit the training data, and one must be selected via an inductive bias. Unfortunately, in direct conflict with the stiff and

nearly-discontinuous behaviors of frictional contact, deep learning techniques tend to select the smoothest interpolator of the data. This behavior is both a predisposition of common training techniques (Belkin et al., 2019; Ribeiro et al., 2020) and an explicit goal of common regularizers such as weight-decay and spectral normalization (Zhang et al., 2019; Miyato et al., 2018). Furthermore, the high-dimensional setting does not allow tractable application of advanced optimization techniques; instead, the domain is limited to first-order stochastic gradient descent (SGD).

In consideration of these challenges, this thesis instead returns to physics based modeling, with the broad goals of 1) scaling the identification problem to the high-dimensional setting; and 2) capturing and computing the extent to which low-fidelity physics models are inherently incapable of making precise predictions. We focus on the domain of rigid-body models with rigid contact, in which extreme sensitivity in the infinite stiffness limit is represented as constrained numerical optimization embedded into the systems dynamics. We take inspiration from many recent robotics algorithms which leverage this optimization structure for effective and efficient simulation, estimation, planning and control (Stewart and Trinkle, 1996; Aydinoglu et al., 2022, 2023; Posa et al., 2013, 2014; Wensing et al., 2022). By careful treatment of the constraints embedded in these dynamical systems, we will discover how even infinitely-stiff dynamics can be identified by optimizing a non-stiff objective (Bianchini, Halm, Matni, and Posa, 2022). With a well-condition optimization technique, we then effectively use DNN's as a computational tool in a *white-box* modeling setting, where they can be used to represent unknown geometries that are reconstructed with minimal priors. For such models with well-identified parameters, we then consider how the simplified rigid-body view of collisions lacks fidelity to correctly predict the outcomes of nearly-simultaneous collisions commonly encountered in robotics—such as heel and toe strikes during a foot step. We develop a theoretical basis to capture partial knowledge of such events as uncertain set-valued outcomes, and again use numerical optimization to compute approximations of such sets.

1.1. Outline and Contributions

We begin in Chapter 2 by presenting a comprehensive background on dynamical systems; rigid-body dynamics; and system identification. In the Chapter 2 and main body of the text, sufficient

background is provided to illuminate the meaning and importance of our primary theoretical and practical results. In the interest of readability, we sequester detailed proofs and technical lemmas to the appendicies, with supplemental theoretical background in Appendix A.

In Chapter 3, we summarize recent and seminal works related to system identification; integration of neural networks into physical models; and neural representations of geometry.

Our primary contributions begin in Chapter 4, in which we present concrete, empirical evidence that naïve deep-learned approaches fail to capture numerically stiff contact behaviors, first published Parmar*, Halm*, and Posa (2021). We isolate the effects of contact's non-smoothness by varying the mechanical stiffness of a compliant contact simulator, and describe and motivate the selection of a canonical "die roll" system upon which such effects are exhaustively evaluated. Even with meticulously optimized hyperparameterd, we find that stiffness alone dramatically degrades the model-fitting processes; generalization, and data-efficiency. Additional experimental details are provided in Appendix B.

In Chapter 5, we leverage the optimization-based structure of rigid-body models to develop an alternative, mechanics-based model-fitting objective to prediction error; this work was originally published in Pfrommer*, Halm*, and Posa (2020) and Bianchini, Halm, Matni, and Posa (2022). We prove that this objective tightly approximates graph distance, a well-accepted but computationally-intractable measure of model quality (Cifarelli, 1988; Adcock, 1878). We demonstrate our objective's superior ability to avoid local minima during the model optimization process. Technical lemmas and proofs are provided in Appendix C.

In Chapter 6, we investigate how physics-based model identification can be scaled from fitting a few scalar parameters to identifying infinite-dimensional unknowns, such as object geometries. In Pfrommer*, Halm*, and Posa (2020), we develop a novel, implicit representation that we call *ContactNets*, in which deep neural networks parameterize geometries as unions of convex shapes. The parameters of these networks can be fit to data by optimizing our mechanics-based objective. We demonstrate that *ContactNets* can identify geometric and frictional properties of an object

rolling, pivoting, and sliding against its environment from less than a minute of motion capture data. Theoretical results are detailed in Appendix D.

Finally in Chapter 7, we examine how, given well-identified parameters, rigid-body models can represent extreme sensitivity to initial conditions with set-valued predictions. We focus on sensitivity due to the rapid ordering or sequencing of impact forces in nearly-simultaneous collisions. As existing rigid-body models lack the fidelity and accuracy to predict this ordering, we develop a set-valued rigid-body model which captures the set of all impact orderings as a continuous-time differential inclusion, published in Halm and Posa (2019) and Halm and Posa (2023). We rigorously prove key theoretical properties of this model, including existence of solutions, and develop a sampling-based algorithm to tightly approximate the set of possible collision outcomes. We demonstrate our approach on several examples drawn from robotic manipulation and locomotion. Both experimental details and theoretical proofs are provided in Appendix E.

We conclude in Chapter 8 with a summary of both the described work and potential future avenues of study.

CHAPTER 2

Background

In this chapter, we discuss the mathematical and mechanical context around our work. We begin with some essential mathematical definitions (Section 2.1); common operations' notation is listed in Table 2.1. We then build a context for physical dynamical models, with an overview of dynamical systems (Section 2.2); set-valued functions (Section 2.3) and dynamics (Section 2.4); rigid-body mechanics (Section 2.5); and dynamical systems with embedded optimization problems (Section 2.6). We summarized associated notation in (Table 2.2). We then turn towards review of essential algorithms for robotics. We define the system identification problem (Section 2.7), and provide additional details for data-driven, neural network modeling in particular (Section 2.8); these concepts will be essential in Chapters 4 to 6. We conclude with some brief tools from sampling-based approximation (Section 2.9), necessary for computational approximations of the set-valued dynamics we derive in Chapter 7.

2.1. Mathematical Definitions

The total derivative of an absolutely continuous function $\boldsymbol{f}(t)$ is denoted $\dot{\boldsymbol{f}}(t)$. $f : A \to B$ is *Lipschitz continuous* with constant L if for all a_1, a_2 in A, $\|f(a_1) - f(a_2)\|_2 \leq L \|a_1 - a_2\|_2$. An absolutely continuous $\boldsymbol{f}(t)$ has this property if $\left\|\dot{\boldsymbol{f}}(t)\right\|_2 \leq L$ almost everywhere (a.e.). The composition of Lipschitz functions is also Lipschitz:

Proposition 1. *let $f, g : A \times B \to A$ be two Lipschitz functions with constants L_f and L_g. Then $h(a, b_1, b_2) = f(g(a, b_1), b_2)$ is Lipschitz with constant $L_f L_g$.*

We say a function $\alpha(s) : \Omega \to \mathrm{cl}\mathbb{R}^+$, is positive definite if it is positive on $\Omega \setminus \{0\}$ and $\alpha(0) = 0$.

2.2. Modeling Robots as Dynamical Systems

Many robotics algorithms—including planners, controllers, estimators, and simulators—are fundamentally *model-based*, in that a core dependency is a dynamical-system model of robots and their

Table 2.1: Frequently-used constants and operations on sets A, A_i, A', B, scalars c, vectors $\boldsymbol{v}, \boldsymbol{w}$, matrices $\boldsymbol{M}, \boldsymbol{N}$, and functions $f : A \to B$, $\boldsymbol{g}(t) : \mathbb{R} \to \mathbb{R}^n$, $D : A \to \mathbb{P}(B)$. For brevity, we frequently write a singleton $\{a\}$ without braces.

Expression	Meaning
A^c	complement of A
$\text{int}(A)$	interior of A
$\text{cl}(A)$	closure of A
$\text{co}(A)$	convex hull of A
$\mathbb{P}(A)$	power set of A
$\boldsymbol{M}A$	scaled set, $\{\boldsymbol{M}a : a \in A\}$
$-A$	$(-1)A$
$A + B$	Minkowski sum of A and B
$A - B$	Minkowski sum of A and $-B$
$[A_1; \ \ldots \ A_k]$	Cartesian product $A_1 \times \cdots \times A_k$
$f(A')$	image of A', $\{f(a') : a' \in A'\}$
$D(A')$	image of A', $\cup_{a' \in A'} D(a')$
$\dot{\boldsymbol{g}}(s)$	total derivative $\frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{g}$
$\boldsymbol{v}_i$	ith element of $\boldsymbol{v}$
$\sigma_{max}(\boldsymbol{M})$	maximum singular value of $\boldsymbol{M}$
$\sigma_{min}(\boldsymbol{M})$	minimum singular value of $\boldsymbol{M}$
$\boldsymbol{M} \succ \boldsymbol{N}$	$\boldsymbol{M} - \boldsymbol{N}$ is positive definite
$\boldsymbol{M} \succeq \boldsymbol{N}$	$\boldsymbol{M} - \boldsymbol{N}$ is pos. semi-definite
$\boldsymbol{v} > \boldsymbol{w}$	$\boldsymbol{v}_i > \boldsymbol{w}_i$ for each i
$\boldsymbol{v} \geq \boldsymbol{w}$	$\boldsymbol{v}_i \geq \boldsymbol{w}_i$ for each i
$A > 0$	each element of A is positive
$A \geq 0$	each element of A is non-negative
$\|\boldsymbol{A}\|_F$	Frobenius norm of A
$\|\boldsymbol{v}\|_p$	l_p norm of $\boldsymbol{v}$ for $p > 0$
$\|\boldsymbol{v}\|_{\boldsymbol{M}}$	$\boldsymbol{M}$-norm $\sqrt{\boldsymbol{v}^T\boldsymbol{M}\boldsymbol{v}}$, $\boldsymbol{M} \succeq \boldsymbol{0}$
$\widehat{\boldsymbol{v}}$	unit direction, $\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|_2}$, of $\boldsymbol{v} \neq \boldsymbol{0}$
$\text{Ball}(c)$	c-radius ball $\{\boldsymbol{v} : \|\boldsymbol{v}\|_2 < c\}$
$\boldsymbol{1}$	matrix/vector of all 1's
$\boldsymbol{0}$	matrix/vector of all 0's
$\mathbb{R}^{n+}$	$\{\boldsymbol{v} \in \mathbb{R}^n, \boldsymbol{v} \geq \boldsymbol{0}\}$
$\text{dual}(A)$	dual cone of A
$\text{proj}(\boldsymbol{v}, A)$	projection of $\boldsymbol{v}$ onto A

Table 2.2: Dynamics terms for rigid bodies and frictional contact. Some terms are written with the dependence on their inputs suppressed.

Term	Meaning
n_q	number of configuration variables
n_v	number of generalized velocities
m	number of contacts
n_θ	number of system parameters
t	time
$\boldsymbol{q}$	robot/environment configuration
$\boldsymbol{v}$	robot/environment velocity
$\boldsymbol{x}$	robot/environment state
$\boldsymbol{x}'$	next robot/environment state
$\boldsymbol{\theta}$	system parameters
$\boldsymbol{u}$	robot/environment input forces
$\boldsymbol{\Gamma}(\boldsymbol{q})$	generalized velocity Jacobian (2.2)
$\boldsymbol{f_\theta}(\boldsymbol{x}, \boldsymbol{u})$	State update equation (2.3)
$\boldsymbol{M}(\boldsymbol{q})$	generalized mass-inertia matrix (2.8)
$F_s(\boldsymbol{x}, \boldsymbol{u})$	non-contact forces (2.8)
$\boldsymbol{\phi}(\boldsymbol{q})$	signed distance between bodies
$K(\boldsymbol{q}, \boldsymbol{v})$	total kinetic energy (2.9)
$\boldsymbol{J}_n(\boldsymbol{q})$	normal velocity Jacobian
$\boldsymbol{J}_t(\boldsymbol{q})$	tangent velocity Jacobian
$\boldsymbol{J}(\boldsymbol{q})$	full contact velocity Jacobian (2.24)
$\mathcal{I}$	set of all contacts
$\mathcal{I}_A(\boldsymbol{q})$	active/touching contact set at $\boldsymbol{q}$ (2.11)
$\mathcal{I}_P(\boldsymbol{q})$	penetrating contact set at $\boldsymbol{q}$ (2.12)
$\mathcal{Q}_A$	set of active-contact configurations
$\mathcal{Q}_P$	set of penetrating configurations
$\bar{\mathcal{X}}_A$	set of active-contact states
$\bar{\mathcal{X}}_P$	set of penetrating states
$\mathcal{C}(\boldsymbol{q})$	set of colliding velocities at $\boldsymbol{q}$ (2.26)
$\mathcal{S}(\boldsymbol{q})$	set of separating velocities at $\boldsymbol{q}$ (2.27)
$\boldsymbol{\lambda}_n$	normal forces vector
$\boldsymbol{\lambda}_t$	frictional contact forces vector
$\boldsymbol{\lambda}$	full contact forces vector (2.25)
$\boldsymbol{\mu}_i$	ith contact Coulomb friction coeff.
$\text{FC}(\boldsymbol{q})$	Coulomb friction cone at $\boldsymbol{q}$ (2.16)
$\text{LFC}(\boldsymbol{q})$	linear friction cone at $\boldsymbol{q}$ (2.20)
$\boldsymbol{J}_D$	linear tangent vel. Jacobian (2.32)
$\boldsymbol{\lambda}_D$	linear friction forces vector (2.32)
$\bar{\boldsymbol{J}}$	linear velocity Jacobian (2.33)
$\bar{\boldsymbol{\lambda}}$	linear contact forces vector (2.33)
$\boldsymbol{\Lambda}_n$	normal impulses vector
$\boldsymbol{\Lambda}_t$	frictional impulses vector
$\boldsymbol{\Lambda}$	full contact impulse vector
$\boldsymbol{g_\theta}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\Lambda})$	optimization-based update (2.37)
$h_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}', \boldsymbol{\Lambda})$	optimization-based objective (2.37)

environments. Continuous-time models are often described with a differential equation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \,. \tag{2.1}$$

Here, the state $\boldsymbol{x} = [\boldsymbol{q}; \boldsymbol{v}]$ captures positions and velocities of the robot and various objects in it's environment via a configuration $\boldsymbol{q} \in \mathbb{R}^{n_q}$ and generalized velocity $\boldsymbol{v}$. Though $\boldsymbol{v} \in \mathbb{R}^{n_v}$ is equal to $\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t}$ for some systems, others (e.g. those relating angular velocities and quaternion derivatives) obey

$$\mathrm{d}\boldsymbol{q} = \boldsymbol{\Gamma}(\boldsymbol{q})\boldsymbol{v}\mathrm{d}t \,, \tag{2.2}$$

for some smooth, bounded, full-column-rank $\boldsymbol{\Gamma}(\boldsymbol{q})$. $\dot{\boldsymbol{x}} = \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}$ represents the time derivative of the state. $\boldsymbol{u} \in \mathbb{R}^{n_u}$ are inputs, such as motor torques, under direct control of the robot. When simulating such dynamics, it is common practice to develop a discrete-time approximation of (2.1),

$$\boldsymbol{x}' = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}) \,, \tag{2.3}$$

where $(\boldsymbol{x}, \boldsymbol{u})$ is the current state and input and $\boldsymbol{x}'$ the next state after a time-step of duration Δt.

The behavior of such systems are defined jointly by their parameters $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ and laws of motion $\boldsymbol{f}$. In the case of a rigid-body model, $\boldsymbol{\theta}$ for instance may include masses; link lengths; and object geometries, while $\boldsymbol{f}$ is some form of Newton's laws of motion. By contrast, some data-driven models, such as neural networks, can have millions of parameters; $\boldsymbol{f}_{\boldsymbol{\theta}}$ in this setting can tightly approximate *any* function, depending on the particular choice of $\boldsymbol{\theta}$.

2.3. Set-Valued Maps

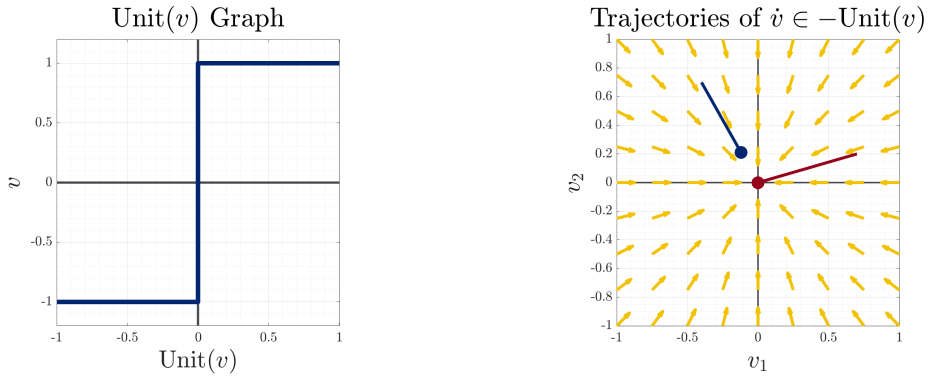
Our mathematical constructions and theoretical results will frequently make use of *set-valued* maps $D(a) : A \to \mathbb{P}(B)$, which take as input an element $a \in A$ an output a subset of B' of some output space B. For notational compactness, we will often use various shorthand forms of set-valued operations, listed as part of Table 2.1. Set-valued maps may exhibit properties reminiscent of continuity for single-valued functions. We in particular will make frequent use of an *upper semi-*

continuity (u.s.c.) property:

Definition 1. *A function $D : A \to \mathbb{P}(B)$ is upper semi-continuous if for any input a and neighborhood B' of $D(a)$, there exists a neighborhood A' of a with $B' \subseteq D(A')$. Equivalently, if B is compact, for all convergent sequences $\{a_n\}_{n\in\mathbb{N}}$ and $\{b_n\}_{n\in\mathbb{N}}$,*

$$b_n \in D(a_n)\,, \forall n \implies \lim b_n \in D(\lim a_n)\,.$$

2.4. Differential Inclusions



(a) (b)

Figure 2.1: (a) Graph of Unit($\boldsymbol{v}$) for $n = 1$. Unit($\boldsymbol{v}$) is continuous at $\boldsymbol{v} \neq 0$. At $\mathbf{0}$, Unit takes the value $[-1, 1]$, which contains a continuous extension of $\hat{\boldsymbol{v}}$ from both the left (-1) and the right ($+1$), so that Unit is u.s.c.. (b) Flow field (yellow) of the solutions (blue, red) to $\dot{\boldsymbol{v}} \in -\text{Unit}(\boldsymbol{v})$ for $\boldsymbol{v} \in \mathbb{R}^2$.

We will later see that in continuous time, the dynamics of rigid bodies under frictional contact present complexities that the ODE formulation (2.1) cannot capture, as multiple outcomes that obey Coulomb's law of friction may exist (*non-unique* behaviors) (Stewart, 2000). It is then useful to define an object that, unlike ODEs, allows for the derivative at each state to lie in a set of possible values

$$\dot{\boldsymbol{x}} \in D(\boldsymbol{x})\,. \tag{2.4}$$

As the set-valued map $D(\boldsymbol{x})$ associated with friction may not be continuous, conditions for a function $\boldsymbol{x}(t)$ to solve this *differential inclusion* (DI) are weaker from those for an ODE:

Definition 2. *For a compact interval* $[a, b]$, $\boldsymbol{x} : [a, b] \to \mathbb{R}^n$ *is a solution to the differential inclusion* $\dot{\boldsymbol{x}} \in D(\boldsymbol{x})$ *if* $\boldsymbol{v}$ *is absolutely continuous and* $\dot{\boldsymbol{x}}(t) \in D(\boldsymbol{x}(t))$ *a.e. on* $[a, b]$. *Denote the set of such solutions as* $\mathrm{SOL}\,(D, [a, b])$.

Solutions to initial value problems for (2.4) are defined similarly:

Definition 3. *The set of solutions to* $\dot{\boldsymbol{x}}(t) \in D(t)$ *with initial condition* $\boldsymbol{x}(a) = \boldsymbol{x}_0$ *over the interval* $t \in [a, b]$ *are denoted as* $\mathrm{IVP}\,(D, \boldsymbol{x}_0, [a, b])$.

In Figure 2.1, we consider an example DI

$$\dot{\boldsymbol{v}} \in -\mathrm{Unit}\,(\boldsymbol{v})\ , \tag{2.5}$$

where $\mathrm{Unit}\,(\boldsymbol{x})$ is the set-valued unit direction function

$$\mathrm{Unit}\,(\boldsymbol{v}) = \begin{cases} \{\widehat{\boldsymbol{v}}\} & \boldsymbol{x} \neq \boldsymbol{0}\,, \\ \{\boldsymbol{v} : \|\boldsymbol{v}\|_2 \leq 1\} = \mathrm{cl}(\mathrm{Ball}(1)) & \boldsymbol{v} = \boldsymbol{0}\,. \end{cases} \tag{2.6}$$

The unique solution to the initial value problem starting from $\boldsymbol{v}(0) = \boldsymbol{v}_0$ has the form

$$\boldsymbol{v}(t) = \max\left(\|\boldsymbol{v}_0\|_2 - t, 0\right) \hat{\boldsymbol{v}}_0\,. \tag{2.7}$$

This solution is non-differentiable at $t = \|\boldsymbol{v}_0\|_2$ and thus is not a solution of any ODE. In general, non-emptiness and regularity of the initial value problem depends on the structure of $D(\boldsymbol{x})$; fortunately, we will later show that solution sets for frictional dynamics are well-behaved due to their upper semi-continuous (u.s.c.) structure:

Proposition 2 (Aubin and Cellina (1984))**.** *Let* $\boldsymbol{x}_0 \in \mathbb{R}^n$ *and* $[a, b]$ *be a compact interval. Suppose* $D(\boldsymbol{x})$ *is uniformly bounded (i.e.* $D(\boldsymbol{x}) \subseteq \mathrm{Ball}(c)$ *for some* $c > 0$*). If* $D(\boldsymbol{x})$ *is u.s.c., closed, convex, and non-empty at all* $\boldsymbol{x}$*, then* $\mathrm{IVP}\,(D, \boldsymbol{x}_0, [a, b])$ *is non-empty and u.s.c. in* $\boldsymbol{x}_0$ *under uniform convergence.*

U.s.c. functions have the useful property that they map compact sets to closed sets, and Proposition 2 immediately and crucially implies that $\mathrm{SOL}\,(D, [a, b])$ and $\mathrm{IVP}\,(D, \boldsymbol{x}_0, [a, b])$ are non-empty and closed under uniform convergence. The DI in Figure 2.1 for example exhibits this structure.

Similar to continuous functions, there are several useful compositional rules which preserve upper semicontinuity; finite combination of u.s.c. functions by cartesian product, convex hull, composition, union, and addition are all u.s.c..

2.5. Rigid-Body Dynamics with Frictional Contact

A popular physics-based model class for robotics is articulated rigid-body models. In these models, bodies do not deform, and their states are completely described by the position; orientation; linear velocity; and angular velocity of a nominal frame affixed to the body.

2.5.1. Continuous-time Evolution Without Impacts

We begin be describing the dynamics of rigid-bodies which may experience finite contact forces under sustained contact, but do not experience collisions or sudden changes in velocity. Contact between these bodies is modeled as occurring at up to m point pairs (for a thorough introduction, see Brogliato (1999) and Stewart (2000)) referred to as the *contacts* $\mathcal{I} = \{1, \ldots, m\}$. Impactless evolution of the system is governed by the *manipulator equations*

$$\boldsymbol{M}(\boldsymbol{q})\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} = F_s(\boldsymbol{x}, \boldsymbol{u}) + \sum_{i \in \mathcal{I}} \boldsymbol{J}_i(\boldsymbol{q})^T \boldsymbol{\lambda}_i \,. \tag{2.8}$$

Here, the continuous function $\boldsymbol{M}(\boldsymbol{q}) \succ 0$ is the generalized inertial matrix, related to the kinetic energy $K(\boldsymbol{q}, \boldsymbol{v})$ by

$$K(\boldsymbol{q}, \boldsymbol{v}) = \frac{1}{2}\left\|\boldsymbol{v}\right\|^2_{\boldsymbol{M}(\boldsymbol{q})} = \frac{1}{2}\boldsymbol{v}^T \boldsymbol{M}(\boldsymbol{q})\boldsymbol{v} \,. \tag{2.9}$$

By assumption, there exist global $c_1, c_2 > 0$ such that $c_1 \boldsymbol{I} \succeq \boldsymbol{M} \succeq c_2 \boldsymbol{I}$. F_s aggregates smooth, non-contact forces (e.g. potential, gyroscopic, and input forces as well as Coriolis and centrifugal effects). For each i, $\boldsymbol{J}_i^T \boldsymbol{\lambda}_i$ is the net (generalized) force due to the ith contact. $\boldsymbol{J}_i = [\boldsymbol{J}_{n,i}; \boldsymbol{J}_{t,i}]$ is the contact Jacobian which maps generalized velocities into Euclidean velocities in the ith contact frame normal ($\boldsymbol{J}_{n,i}$) and tangential ($\boldsymbol{J}_{t,i}$) directions. $\boldsymbol{\lambda}_i = [\boldsymbol{\lambda}_{n,i}; \boldsymbol{\lambda}_{t,i}]$ are the contact-frame normal

forces $\boldsymbol{\lambda}_{n,i}$ and frictional forces $\boldsymbol{\lambda}_{t,i}$, which are typically dictated by two essential physical laws:

- **Normal complementarity**: The signed distance $\boldsymbol{\phi}(\boldsymbol{q}) \in \mathbb{R}^m$ captures object geometry as inter-body distances. Normal forces push bodies apart, and neither penetration nor force-at-a-distance are possible; that is, for each i,

$$\boldsymbol{J}_{n,i} = \frac{\partial \boldsymbol{\phi}_i}{\partial \boldsymbol{q}} \boldsymbol{\Gamma}\,, \qquad\qquad \mathbf{0} \le \boldsymbol{\lambda}_{n,i} \perp \boldsymbol{\phi}_i(\boldsymbol{q}) \ge \mathbf{0}\,. \tag{2.10}$$

  We denote the active and penetrating contacts at $\boldsymbol{q}$ as

$$\mathcal{I}_A(\boldsymbol{q}) = \{i \in \mathcal{I} : \boldsymbol{\phi}_i(\boldsymbol{q}) \le 0\}\,, \tag{2.11}$$

$$\mathcal{I}_P(\boldsymbol{q}) = \{i \in \mathcal{I} : \boldsymbol{\phi}_i(\boldsymbol{q}) < 0\}\,. \tag{2.12}$$

- **Maximal dissipation**: Friction dissipates as much power ($\boldsymbol{J}_{t,i}\boldsymbol{v} \cdot \boldsymbol{\lambda}_{t,i}$) as possible. Couloub friction with coefficient $\boldsymbol{\mu}_i$ in particular obeys this property within the admissible set

$$\left\{\boldsymbol{\lambda}_{t,i} : \|\boldsymbol{\lambda}_{t,i}\|_2 \le \boldsymbol{\mu}_i \boldsymbol{\lambda}_{n,i}\right\}\,. \tag{2.13}$$

  The corresponding set of generalized forces is the *friction cone*

$$\mathrm{FC}(\boldsymbol{q}) = \sum_{i \in \mathcal{I}_A(\boldsymbol{q})} \left\{\boldsymbol{J}_i(\boldsymbol{q})^T \boldsymbol{\lambda}_i : \|\boldsymbol{\lambda}_{t,i}\|_2 \le \boldsymbol{\mu}_i \boldsymbol{\lambda}_{n,i}\right\}\,.$$

  The maximally-dissipative friction force and associated generalized force opposes the sliding direction $\mathrm{Unit}(\boldsymbol{J}_{t,i}\boldsymbol{v})$ as much as possible:

$$\boldsymbol{\lambda}_{t,i} \in -\boldsymbol{\mu}_i \boldsymbol{\lambda}_{n,i} \mathrm{Unit}(\boldsymbol{J}_{t,i}\boldsymbol{v})\,, \tag{2.14}$$

$$F_i(\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\lambda}_{n,i}) = \left(\boldsymbol{J}_{n,i}^T - \boldsymbol{\mu}_i \boldsymbol{J}_{t,i}^T \mathrm{Unit}(\boldsymbol{J}_{t,i}\boldsymbol{v})\right) \boldsymbol{\lambda}_{n,i}\,. \tag{2.15}$$

We note in particular the identity

$$\mathrm{FC}\left(\boldsymbol{q}\right) = \sum_{i \in \mathcal{I}_A(\boldsymbol{q})} F_i(\boldsymbol{q}, 0, \mathbb{R}^+)\,. \tag{2.16}$$

A common variant of this model is the *linearized* Coulomb model, in which the admissible set is replaced with $\{\boldsymbol{\lambda}_{t,i} \in \boldsymbol{\mu}_i \boldsymbol{\lambda}_{n,i} \mathrm{co}\left(\{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_k\}\right)\}$ for some unit-length vectors $\boldsymbol{D} = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_k]$, leading to similar definitions of forces and a *linearized friction cone*:

$$\mathrm{Unit}_{\boldsymbol{D}}(\boldsymbol{r}) = \mathrm{co}\left(\arg\max_{d_i} \boldsymbol{d}_i \cdot \boldsymbol{r}\right)\,, \tag{2.17}$$

$$\boldsymbol{\lambda}_{t,i} \in -\boldsymbol{\mu}_i \boldsymbol{\lambda}_{n,i} \mathrm{Unit}_{\boldsymbol{D}}(\boldsymbol{J}_{t,i}\boldsymbol{v})\,, \tag{2.18}$$

$$F_{\boldsymbol{D},i}(\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\lambda}_{n,i}) = \left(\boldsymbol{J}_{n,i}^T - \boldsymbol{\mu}_i \boldsymbol{J}_{t,i}^T \mathrm{Unit}_{\boldsymbol{D}}(\boldsymbol{J}_{t,i}\boldsymbol{v})\right) \boldsymbol{\lambda}_{n,i}\,, \tag{2.19}$$

$$\mathrm{LFC}\left(\boldsymbol{q}\right) = \sum_{i \in \mathcal{I}_A(\boldsymbol{q})} F_{\boldsymbol{D},i}(\boldsymbol{q}, 0, \mathbb{R}^+)\,. \tag{2.20}$$

The identity $\mathrm{Unit}_{\boldsymbol{D}}(\boldsymbol{r}) \subseteq \mathrm{Unit}_{\boldsymbol{D}}(\boldsymbol{0}) \subseteq \mathrm{Unit}(\boldsymbol{0})$ leads to

$$\textstyle\sum_{i \in \mathcal{I}_A(\boldsymbol{q})} F_{\boldsymbol{D},i}(\boldsymbol{q}, \boldsymbol{v}, \mathbb{R}^+) \subseteq \mathrm{LFC}\left(\boldsymbol{q}\right) \subseteq \mathrm{FC}\left(\boldsymbol{q}\right)\,. \tag{2.21}$$

$\boldsymbol{\phi}$ is Lipschitz and continuously differentiable. We also assume that for all active, non-penetraing contacts, there exists a generalized velocity for which the contact is separating:

Assumption 1. $\forall i,\ \boldsymbol{\phi}_i(\boldsymbol{q}) = 0 \implies \boldsymbol{J}_{n,i}(\boldsymbol{q}) \neq \boldsymbol{0}$.

$\boldsymbol{J}_{n,i}$ is bounded and continuous by the properties of $\boldsymbol{\phi}$ and $\boldsymbol{\Gamma}$, while $\boldsymbol{J}_{t,i}$ has the same properties by assumption. These properties can be guaranteed, for instance, for piecewise-smooth bodies with bounded curvature. We note that because $\boldsymbol{\phi}$ is continuous, $\mathcal{I}_A(\boldsymbol{q})$ and $\mathcal{I} \setminus \mathcal{I}_P(\boldsymbol{q})$ are u.s.c. in $\boldsymbol{q}$. From these functions we also define $\mathcal{Q}_A = \{\boldsymbol{q} : \mathcal{I}_A(\boldsymbol{q}) \neq \emptyset\}$, the configurations with active contact, and $\mathcal{Q}_P = \{\boldsymbol{q} : \mathcal{I}_P(\boldsymbol{q}) \neq \emptyset\}$, the interpenetrating configurations.

We will often see that various theoretical guarantees (seminally including existence of solutions in

continuous and discrete time (Stewart, 2000)) for such systems depend on a *pointedness* assumption on the friction cone FC:

Assumption 2 (Pointed Friction Cone)**.** *At any configuration* $\boldsymbol{q}$*, the friction cone* $\mathrm{FC}(\boldsymbol{q})$ *is pointed in some direction* $\boldsymbol{d}(\boldsymbol{q})$*:*

$$\forall \boldsymbol{F} \in \mathrm{FC}(\boldsymbol{q}), \boldsymbol{d}(\boldsymbol{q}) \cdot \boldsymbol{F} \geq \|\boldsymbol{F}\|_2 \ . \tag{2.22}$$

Therefore, there also exists $p(\boldsymbol{q})$ *such that for any* $\boldsymbol{\lambda}$ *in the Coulomb admissible set* (2.13)*,*

$$\left\|\boldsymbol{J}^T \boldsymbol{\lambda}\right\|_2 \geq p(\boldsymbol{q}) \|\boldsymbol{\lambda}\|_2 \ . \tag{2.23}$$

Finally, we define the following notation:

$$\boldsymbol{J}_n = \begin{bmatrix} \boldsymbol{J}_{n,1} \\ \vdots \\ \boldsymbol{J}_{n,m} \end{bmatrix}, \quad \boldsymbol{J}_t = \begin{bmatrix} \boldsymbol{J}_{t,1} \\ \vdots \\ \boldsymbol{J}_{t,m} \end{bmatrix}, \quad \boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_n \\ \boldsymbol{J}_t \end{bmatrix}, \tag{2.24}$$

$$\boldsymbol{\lambda}_n = \begin{bmatrix} \boldsymbol{\lambda}_{n,1} \\ \vdots \\ \boldsymbol{\lambda}_{n,m} \end{bmatrix}, \quad \boldsymbol{\lambda}_t = \begin{bmatrix} \boldsymbol{\lambda}_{t,1} \\ \vdots \\ \boldsymbol{\lambda}_{t,m} \end{bmatrix}, \quad \boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_n \\ \boldsymbol{\lambda}_t \end{bmatrix}, \tag{2.25}$$

$$\mathcal{C}(\boldsymbol{q}) = \{\boldsymbol{v} \in \mathbb{R}^{n_v} : \exists i \in \mathcal{I}_A(\boldsymbol{q}),\, \boldsymbol{J}_{n,i}\boldsymbol{v} < 0\}, \tag{2.26}$$

$$\mathcal{S}(\boldsymbol{q}) = \{\boldsymbol{v} \in \mathbb{R}^{n_v} : \forall i \in \mathcal{I}_A(\boldsymbol{q}),\, \boldsymbol{J}_{n,i}\boldsymbol{v} > 0\}. \tag{2.27}$$

$\mathcal{C}(\boldsymbol{q})$ is the set of *colliding* velocities, for which an active contact is moving towards penetration and must cause an impact. $\mathcal{S}(\boldsymbol{q})$ is the set of *separating* velocities, where no impact occurs as all contacting surfaces are moving away from each other. While $\mathcal{C}(\boldsymbol{q})$ and $\mathcal{S}(\boldsymbol{q})$ are disjoint, there may be some velocities in *neither* set; these cases may generate impacts, as in Painlevé's Paradox (Stewart, 2000). By Assumption 1, when $\boldsymbol{q}$ is non-penetrating, $\mathcal{S}(\boldsymbol{q}) = \mathrm{int}\left(\mathcal{C}(\boldsymbol{q})^c\right)$ and $\mathcal{C}(\boldsymbol{q}) = \mathrm{int}\left(\mathcal{S}(\boldsymbol{q})^c\right)$.

2.5.2. Instantaneous, Inelastic Impact Laws

(2.8), (2.10), and (2.14) provide only a partial solution to initial value problems (IVPs). Bodies can collide or come into contact with non-zero velocity ($\boldsymbol{\phi}_i(t) = 0$ and $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\phi}_i(t) < 0$); penetration therefore must be avoided via an *impact* or instantaneous velocity jump

$$\boldsymbol{M}(\boldsymbol{q})(\boldsymbol{v}^+(t) - \boldsymbol{v}^-(t)) = \textstyle\sum_{i \in \mathcal{I}_A(\boldsymbol{q})} \boldsymbol{J}_i(\boldsymbol{q})^T \boldsymbol{\Lambda}_i, \tag{2.28}$$

arising from instantaneous contact *impulses* $\boldsymbol{\Lambda}_i$. As $\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t}$ does not exist, an alternative formulation to ODEs equations in time is required to capture this behavior.

Several models select $\boldsymbol{\Lambda}$ via an impulsive analog to Coulomb's friction law (Anitescu and Potra, 1997; Glocker and Pfeiffer, 1995; Routh, 1891), with additional constraints pertaining to the elasticity of the impact. We focus discussion and our own modeling efforts on *inelastic* collisions, which are well defined in the single-impact case via the constraint $\boldsymbol{J}_{n,i}\boldsymbol{v}^+ = 0$. Each discussed model makes its own generalization of this concept to simultaneous impacts, and there is in general no single accepted rule (Stewart, 2000). In Chapter 7 we will consider and combine concepts from two families of such models: algebraic and differential. We will discuss how each method makes its own nuanced translations of the complementarity and maximal dissipation laws for sustained contact, resulting in distinct theoretical and computational characteristics.

Algebraic methods calculate $\boldsymbol{\Lambda}_i$ as the solution to a finite-dimensional system of algebraic equations (Anitescu and Potra, 1997; Hurmuzlu and Marghitu, 1994; Glocker and Pfeiffer, 1995), which relate the pre- and post-impact velocities to the impact's underlying impulses. Such systems of equations can be approximately computed via numerical optimization.

In some of these models, all impacts are resolved simultaneously. For instance, Glocker and Pfeiffer (1995) and Anitescu and Potra (1997) solve for an impulse $\boldsymbol{\Lambda}$ which both prevents penetration and

(approximately) satisfies linearized Coulomb friction at the *post*-impact velocity $\boldsymbol{v}^+$:

$$\begin{aligned}
\text{find} \quad & \boldsymbol{v}^+; \{\boldsymbol{\Lambda}_i : i \in \mathcal{I}_A\} \,, && \text{(2.29a)} \\
\text{s.t.} \quad & \text{impulse balance (2.28)} \,, && \text{(2.29b)} \\
& \mathbf{0} \leq \boldsymbol{\Lambda}_{n,i} \perp \boldsymbol{J}_{n,i}\boldsymbol{v}^+ \geq \mathbf{0} \,, && \text{(2.29c)} \\
& \boldsymbol{\Lambda}_{t,i} \in -\boldsymbol{\mu}_i \boldsymbol{\Lambda}_{n,i} \text{Unit}_{\boldsymbol{D}}(\boldsymbol{J}_{t,i}\boldsymbol{v}^+) \,. && \text{(2.29d)}
\end{aligned}$$

A critical feature of the algebraic formulation (2.29) is the use of linearized Coulomb friction, which allows it to be cast as a linear complementarity problem (LCP); this class of problems is further discussed in Section 2.6. We refer the reader to Stewart and Trinkle (1996) for a full description, but provide a short summary below. Letting $\boldsymbol{\lambda}_{t,i} = \boldsymbol{D}\boldsymbol{\lambda}_{D,i}$ and $\boldsymbol{J}_{D,i} = \boldsymbol{D}^T\boldsymbol{J}_{t,i}$, (2.29d) can be captured as as the complementarity constraints

$$\mathbf{0} \leq \boldsymbol{\lambda}_{D,i} \perp \boldsymbol{J}_{D,i}(\boldsymbol{q}^+)\boldsymbol{v}^+ + \mathbf{1}\gamma_i \geq \mathbf{0} \,, \tag{2.30}$$

$$\mathbf{0} \leq \gamma_i \perp \boldsymbol{\mu}_i \boldsymbol{\lambda}_{n,i} - \mathbf{1}^T \boldsymbol{\lambda}_{D,i} \geq \mathbf{0} \,. \tag{2.31}$$

For convenience, we define the lumped terms

$$\boldsymbol{\lambda}_D = \begin{bmatrix} \boldsymbol{\lambda}_{D,1} \\ \vdots \\ \boldsymbol{\lambda}_{D,m} \end{bmatrix} \,, \qquad \boldsymbol{J}_D = \begin{bmatrix} \boldsymbol{J}_{D,1} \\ \vdots \\ \boldsymbol{J}_{D,m} \end{bmatrix} \,, \tag{2.32}$$

$$\bar{\boldsymbol{\lambda}} = \begin{bmatrix} \boldsymbol{\lambda}_n \\ \boldsymbol{\lambda}_D \end{bmatrix} \,, \qquad \bar{\boldsymbol{J}} = \begin{bmatrix} \boldsymbol{J}_n \\ \boldsymbol{J}_D \end{bmatrix} \,. \tag{2.33}$$

This casting of multiple, simultaneous impacts as a single LCP is a significant computational advantage, as only one, solvable numerical program must be instantiated to calculate the post-impact velocity. However, the constraints embedded in this problem are often violated in real systems with

multiple contacts, in particular the so-called velocity-based complementarity (2.29c) formulation of inelasticity (Chatterjee, 1999).

An alternative algebraic view of simultaneous impacts that does not require the same velocity-based complementarity constraints is to resolve multi-impact as a sequence of individual impacts, as in Ivanov (1995); Smith et al. (2012); Seghete and Murphey (2014); and many other models. To summarize this technique:

1. Pick a single active contact $i \in \mathcal{I}_A(\boldsymbol{q})$.

2. Resolve a single impact at i with some impulse $\boldsymbol{\Lambda}_i$, and increment $\boldsymbol{v} \leftarrow \boldsymbol{v} + \boldsymbol{M}^{-1}\boldsymbol{J}_i^T\boldsymbol{\Lambda}_i$.

3. Terminate and take $\boldsymbol{v}^+ = \boldsymbol{v}$ if it is non-colliding ($\boldsymbol{v} \notin \mathcal{C}(\boldsymbol{q})$); otherwise, return to step 1.

Various methods differ in their choice of contact ordering as will as single-impact resolution, resulting in distinctly different final outcomes to the same initial conditions. Some such methods are only able to guarantee that the process terminates under significant assumptions, e.g. two or fewer contacts (Seghete and Murphey, 2014). In Chapter 7, sequences of single impacts resolved using (2.29) will serve as a point of comparison for a new collision law that we develop. Finally, such methods by design are unable to directly represent partially-concurrent impacts that occur in real-world systems.

As opposed to algebraic models, differential impact models consider continuous evolution of velocity from pre- to post-impact velocity, in which the total derivative of $\boldsymbol{v}$ satisfies laws of frictional contact in some form. In the context of rigid contact models, this derivative $\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}s}$ is with respect to a variable of integration s which does not correspond to time, but rather measures the impulse accumulated over an instantaneous collision. At least in a limited capacity, such methods do directly represent the time-dependence and continual evolution of real-world object velocities during impact, which in Section 7.4 will allow us to represent partially-concurrent impacts resolving at arbitrary relative rates. This fidelity however necessitates computationally expensive simulation of non-smooth or constrained differential equations to resolve impacts, and thus such methods have not been a focus of modern, efficient simulation.

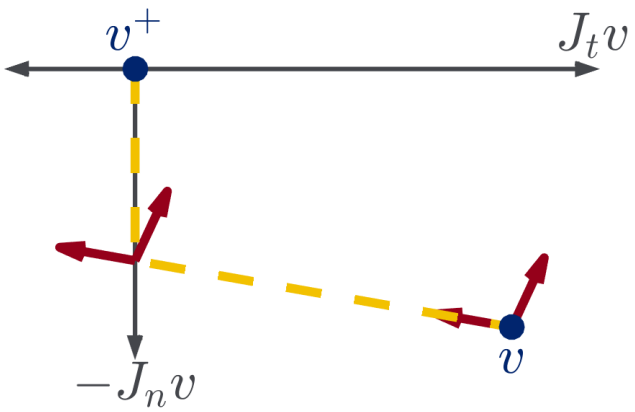


Figure 2.2: Velocity through an impact resolved by Routh's method (adapted from Posa et al. (2016b)). The extreme rays of the friction cone are shown as solid red arrows. The contact begins in a sliding regime. When $\boldsymbol{v}$, shown in the yellow dotted line, intersects $\boldsymbol{J}_t\boldsymbol{v} = \boldsymbol{0}$, the contact transitions to sticking and the impact terminates when $\boldsymbol{J}_n\boldsymbol{v} = \boldsymbol{0}$.

We will now describe one of the oldest differential models for a single impact (Routh, 1891), which we will later extend to the multi-contact case. For a single contact $\mathcal{I}_A(\boldsymbol{q}) = \{i\}$, Routh (1891) proposed a method which satisfies Coulomb friction differentially. To summarize this technique,

1. Increase the normal impulse $\boldsymbol{\Lambda}_{n,i}$ with slope $\boldsymbol{\lambda}_{n,i} = 1$.

2. Increment the tangential impulse with slope $\boldsymbol{\lambda}_{t,i}$ satisfying Coulomb friction (2.14) for the mid-impact velocity $\bar{\boldsymbol{v}} = \boldsymbol{v} + \boldsymbol{M}^{-1}\boldsymbol{J}_i^T\boldsymbol{\Lambda}_i$.

3. Stop at the inelastic condition $\boldsymbol{J}_{n,i}\bar{\boldsymbol{v}} = 0$; set $\boldsymbol{v}^+ = \bar{\boldsymbol{v}}$.

As observed in Posa et al. (2016b), this process is equivalent to the DI

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}s} \in \boldsymbol{M}(\boldsymbol{q})^{-1}F_i(\boldsymbol{q}, \boldsymbol{v}(s), 1)\,. \tag{2.34}$$

Note that for a frictionless contact ($\boldsymbol{\mu} = 0$), this simplifies to $\boldsymbol{M}\dot{\boldsymbol{v}} = \boldsymbol{J}_{n,i}^T$. A diagram depicting the resolution of a planar impact with this method is shown in Figure 2.2. Solutions may transition from sliding to sticking, and the direction of slip may even reverse. While the path is piecewise linear in the planar case, this is not necessarily true in three dimensions. We additionally note that

while (2.34) predicts "forces" even when $\boldsymbol{v}$ is separating ($\boldsymbol{J}_{n,i}\boldsymbol{v} > 0$), Routh's method is by definition only used on velocity trajectories starting with $\boldsymbol{J}_{n,i}\boldsymbol{v} \leq 0$ until the first moment that that $\boldsymbol{J}_{n,i}\boldsymbol{v} = 0$, and thus inelasticity is preserved.

Implicit in Routh's method is an assumption that the terminal condition in step 3) will eventually be reached; if it is possible to get "stuck" with $\boldsymbol{J}_{n,i}\boldsymbol{v} < 0$ forever, then Routh's method would be ill-defined and not predict a post impact state. This does not happen in the frictionless case, as $\boldsymbol{J}_{n,i}\boldsymbol{v}$ has constant positive derivative $\boldsymbol{J}_{n,i}\dot{\boldsymbol{v}} = \|\boldsymbol{J}_{n,i}\|^2_{\boldsymbol{M}^{-1}}$. With more careful treatment capturing kinetic energy dissipation, a similar result can be shown for the frictional case:

Lemma 1 (Single Impact Termination (Appendix E.2.2))**.** *Let $\boldsymbol{q} \notin \mathcal{Q}_P$ be a non-penetrating configuration, and $i \in \mathcal{I}_A(\boldsymbol{q})$ be an active contact. Then there exists $\kappa(\boldsymbol{q}) > 0$ such that for any solution $\boldsymbol{v}(s) : [0, \|\boldsymbol{v}(0)\|_2 \kappa(\boldsymbol{q})] \to \mathbb{R}^{n_v}$ of the single frictional contact system* (2.34)*, $\boldsymbol{v}(s)$ exits the impact at some $s^* \leq \|\boldsymbol{v}(0)\|_2 \kappa(\boldsymbol{q})$; i.e., $\boldsymbol{J}_{n,i}\boldsymbol{v}(s^*) \geq 0$.*

The implication of Lemma 1 is that *a priori*, one can determine an $s > 0$ proportional to the pre-impact speed $\|\boldsymbol{v}\|_2$ such that any solution to the DI (2.34) on $[0, s]$ can be used to construct the post-impact velocity $\boldsymbol{v}^+$. We will see, however, that the extension of this methodology to multiple concurrent impacts is non-trivial, and that physical systems associated with these models often exhibit non-uniqueness.

2.5.3. Initial value problems through impact

Any complete solution to continuous-time IVP's for rigid bodies undergoing impacts must somehow combine the sustained-contact and instantaneous impact models described above. We highlight three properties that such a model would ideally have:

1. **Complete consistency**: Solutions to IVP's should exist from any initial condition.

2. **Full specification**: Essential characterizations of contact behavior should be specifiable, including friction laws or impact (in)elasticity.

3. **Wide scope**: The large majority of multi-body systems should be captured by the framework.

These and similar ideals have been pursued for over a century; see Stewart (2000) for a comprehensive overview. We highlight a few recent and seminal advances towards these properties.

Early hybrid-system formulations generate impacts if and only if a collision occurs. Such laws were determined to lead to non-existence issues under high friction (Jellet, 1872; Painlevé, 1895). Recent advancements in hybrid systems guarantee existence, but instead require all contacting bodies to be nearly-massless (Johnson et al., 2016; Burden et al., 2016). Building on the early ideas of Lecornu (1905), Moreau (1977) developed an alternative measure differential inclusion (MDI) formalism which permits non-zero impulses in $\mathrm{FC}(\boldsymbol{q})$ to occur over an infinitesimal time period $\mathrm{d}t$. However, the preeminent IVP existence proof for general multi-contact MDI's forgoes specification of impact elasticity (Stewart, 1998).

In Section 7.3, we derive a DI model which generally applies to multi-body systems; specifies impacts to be inelastic; and guarantees existence of solutions under similar conditions to Stewart (1998) (see Assumption 2). While DI's have long been used in rigid-body dynamics Leine and van de Wouw (2008), this thesis and recent work (Nurkanović et al., 2021a,b) are the first to solve IVPs through impacts via adding time as a state. This work is the first such DI to capture inelasticity and friction in impact. We will also develop our own discrete-time simulator in Section 7.4, which shares a similar LCP structure to Stewart and Trinkle (1996).

2.5.4. Discrete-time simulation

When simulating the continuous-time dynamics (2.8), (2.2) in discrete-time, it is important to note the interplay between the dynamics, non-penetration constraints, and friction. In particular, many contact forces $\boldsymbol{\lambda}$ may comply with normal complementarity (2.10) and Coulomb friction (2.14) at a given time. However, picking one arbitrarily and simulating (2.8) forward with common ODE integration schemes (e.g. an Euler step) may lead to penetration at the very next time step. Some approaches resolve this issue with a constrained semi-implicit Euler scheme (Stewart and Trinkle, 1996). The essential idea is to integrate the manipulator equations (2.8) to relate a change in velocity to net contact impulses, and to apply the contact constraints (2.10), (2.14) at the end of the time step. In summary, a simulation step maps the current state $\boldsymbol{x} = [\boldsymbol{q}; \boldsymbol{v}]$ and input $\boldsymbol{u}$ to the

next state $\boldsymbol{x}' = [\boldsymbol{q}'; \boldsymbol{v}'] = [\boldsymbol{q} + \Delta\boldsymbol{q}; \boldsymbol{v} + \Delta\boldsymbol{v}]$ by solving

$$\begin{aligned}
\text{find} \quad & \Delta\boldsymbol{q}; \Delta\boldsymbol{v}; \{\boldsymbol{\Lambda}_i : i \in \mathcal{I}_A\}\,, && \text{(2.35a)}\\
\text{s.t.} \quad & \boldsymbol{M}(\boldsymbol{q})\Delta\boldsymbol{v} = F_s(\boldsymbol{x}, \boldsymbol{u})\Delta t + \sum_i \boldsymbol{J}_i^T \boldsymbol{\Lambda}_i\,, && \text{(2.35b)}\\
& \Delta\boldsymbol{q} = \boldsymbol{\Gamma}(\boldsymbol{q})\boldsymbol{v}'\Delta t\,, && \text{(2.35c)}\\
& \boldsymbol{0} \leq \boldsymbol{\Lambda}_{n,i} \perp \boldsymbol{\phi}_i(\boldsymbol{q}) + \frac{\partial \boldsymbol{\phi}_i(\boldsymbol{q})}{\partial \boldsymbol{q}}\Delta\boldsymbol{q} \geq \boldsymbol{0}\,, && \text{(2.35d)}\\
& \boldsymbol{\Lambda}_{t,i} \in -\boldsymbol{\mu}_i \boldsymbol{\Lambda}_{n,i} \text{Unit}(\boldsymbol{J}_{t,i}\boldsymbol{v}')\,. && \text{(2.35e)}
\end{aligned}$$

We note that (2.35d) contains a linearization of the change in interbody signed distance over the timestep:

$$\bar{\boldsymbol{\phi}}_i(\boldsymbol{q}, \boldsymbol{v}') = \boldsymbol{\phi}_i(\boldsymbol{q}) + \frac{\partial \boldsymbol{\phi}_i(\boldsymbol{q})}{\partial \boldsymbol{q}}\Delta\boldsymbol{q} = \boldsymbol{\phi}_i(\boldsymbol{q}) + \boldsymbol{J}_{n,i}\boldsymbol{v}'\Delta t \approx \boldsymbol{\phi}_i(\boldsymbol{q}') \tag{2.36}$$

This numerical program appears quite similar to the instantaneous impact resolution (2.29). Computationally, the methods have similar properties. Due to the linearization in (2.35d), with the linearized linearized coulomb law, Stewart and Trinkle (1996) seminally demonstrated that (2.35) can also be approximately cast as a solvable LCP. However, conceptually, the meaning of $\boldsymbol{\Lambda}_i$ has changed. In (2.29), $\boldsymbol{\Lambda}_i$ is an instantaneous impulse that describes an impact, whereas $\boldsymbol{\Lambda}_i$ in (2.35) represents the total impulse over the time-step, with contributions coming both from sustained contact forces and impacts.

2.6. Optimization-Based Dynamics

We have seen how rigid-body simulation can be formulated as numerical optimization in (2.35). Such models lie in a more general class of *optimization-based* dynamics, in which the mapping from the current state and input to the next state is formulated as an optimization problem over some hidden variables $\boldsymbol{\Lambda}$. For second-order systems, these formulations have the form

$$\begin{aligned}
& \boldsymbol{v}' = \boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\Lambda}^*)\,, && \text{(2.37a)}\\
& \boldsymbol{\Lambda}^* \in \arg\min_{\boldsymbol{\Lambda} \in \mathcal{K}} h_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}', \boldsymbol{\Lambda})\,. && \text{(2.37b)}
\end{aligned}$$

This format decouples the dynamics into a system of equations: one algebraic, describing the state update $\boldsymbol{x}'$ as a function of the hidden variables $\boldsymbol{\Lambda}^*$; and the other expressing constraints relating the hidden variables and the state update as a minimization. In general, to compute forward predictions for such systems, the update function $\boldsymbol{g_\theta}$ is embedded into optimization objective $h_{\boldsymbol{\theta}}$:

$$\boldsymbol{\Lambda}^* \in \arg\min h_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{g_\theta}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\Lambda}), \boldsymbol{\Lambda})\,. \tag{2.38}$$

2.6.1. Rigid-body Simulation as Optimization-based Dynamics

In the particular case of rigid-body simulation (2.35), $\boldsymbol{\Lambda} = [\boldsymbol{\Lambda}_1; \ldots; \boldsymbol{\Lambda}_m]$ represents the contact impulses. The set of admissible impulses $\mathcal{K}$ is given by the Coulomb friction admissible set $\mathcal{K} = L_1 \times \cdots \times L_m$, where $L_i = \left\{\boldsymbol{\Lambda}_i : \|\boldsymbol{\Lambda}_{t,i}\|_2 \leq \boldsymbol{\mu}_i \boldsymbol{\Lambda}_{n,i}\right\}$ is the ($\boldsymbol{\mu}_i$-scaled) Lorentz cone. $\boldsymbol{g_\theta}$ is the manipulator equations:

$$\boldsymbol{g_\theta} = \boldsymbol{v} + \boldsymbol{M}^{-1}\left(F_s \Delta t + \boldsymbol{J}^T \boldsymbol{\Lambda}\right)\,. \tag{2.39}$$

Finally, $h_{\boldsymbol{\theta}}$ represents contact constraints. While there are multiple valid choices of $h_{\boldsymbol{\theta}}$, we note one in particular:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}', \boldsymbol{\Lambda}) = \sum_i \frac{1}{2}\mathrm{neg}(\bar{\boldsymbol{\phi}}_i(\boldsymbol{q}, \boldsymbol{v}'))^2 + \boldsymbol{c}_i(\boldsymbol{q}, \boldsymbol{v}') \cdot \boldsymbol{\Lambda}_i + \boldsymbol{d}_i(\boldsymbol{v}') \cdot \boldsymbol{\Lambda}_i\,, \tag{2.40a}$$

$$\boldsymbol{c}_i(\boldsymbol{q}, \boldsymbol{v}') = \begin{bmatrix} \mathrm{pos}(\bar{\boldsymbol{\phi}}_i(\boldsymbol{q}, \boldsymbol{v}')) \\ \boldsymbol{0} \end{bmatrix}\,, \tag{2.40b}$$

$$\boldsymbol{d}_i(\boldsymbol{v}') = \begin{bmatrix} \|\boldsymbol{J}_{t,i}\boldsymbol{v}'\|_2 \\ \frac{1}{\boldsymbol{\mu}_i}\boldsymbol{J}_{t,i}\boldsymbol{v}' \end{bmatrix}\,. \tag{2.40c}$$

With this construction, (2.38)–(2.39) and (2.35) are equivalent. We highlight that $h_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}', \boldsymbol{\Lambda})$ as described in (2.40) is convex in $\boldsymbol{\Lambda}$, which would suggest that the forward simulation problem is tractable. Unfortunately, as seen in (2.38), during forward prediction, $\boldsymbol{\Lambda}$ also enters into the third argument of h, and it is this joint dependence that can cause the prediction problem to be non-convex and even NP-Hard (Baraff, 1991). We finally note that for most rigid-body systems, the

forward-simulation problem is always solvable Stewart and Trinkle (1996), and thus we may assume that the optimization of $h_{\boldsymbol{\theta}}$ is equivalent to a constraint-satisfaction problem:

Assumption 3. *For the rigid-body dynamical system described in* (2.38)–(2.39)*,*

$$\min_{\boldsymbol{\Lambda}\in\mathcal{K},\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}'} h_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}',\boldsymbol{\Lambda}) = \min_{\boldsymbol{\Lambda}\in\mathcal{K},\boldsymbol{x},\boldsymbol{u}} h_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{u}),\boldsymbol{\Lambda}) = 0\,. \tag{2.41}$$

2.6.2. Solving Forward Simulation with Linear Complementarity

As discussed in Section 2.5.4, forward simulation can be cast as a non-convex linear complementarity problem (LCP). We now briefly define and describe this problem class. We refer the reader to Cottle et al. (2009) for a complete description, and to Stewart and Trinkle (1996) for details on how rigid-body problems are encoded as an LCP.

Definition 4. *The linear complementarity problem with parameters* $\boldsymbol{W}\in\mathbb{R}^{n\times n}$ *and* $\boldsymbol{w}\in\mathbb{R}^n$ *is the constraint satisfaction problem*

$$\begin{aligned} &\textit{find} && \boldsymbol{z}\in\mathbb{R}^n\,, && (2.42)\\ &\textit{subject to} && \boldsymbol{z}^T(\boldsymbol{W}\boldsymbol{z}+\boldsymbol{w}) = \boldsymbol{0}\,, && (2.43)\\ & && \boldsymbol{z},\boldsymbol{W}\boldsymbol{z}+\boldsymbol{w}\geq\boldsymbol{0}\,, && (2.44) \end{aligned}$$

for which the set of solutions is denoted $\mathrm{LCP}(\boldsymbol{W},\boldsymbol{w})$*.* (2.43)–(2.44) *are often abbreviated as* $\boldsymbol{0}\leq\boldsymbol{z}\perp\boldsymbol{W}\boldsymbol{z}+\boldsymbol{w}\geq\boldsymbol{0}$*.*

For LCPs related to frictional behavior, $\boldsymbol{W}$ is often copositive (i.e. $\boldsymbol{x}^T\boldsymbol{W}\boldsymbol{x}\geq 0$ for all $\boldsymbol{x}\geq\boldsymbol{0}$). This property provides a sufficient condition for LCP feasibility and computability:

Proposition 3 (Cottle et al. (2009))**.** *Let* $\boldsymbol{w}\in\mathbb{R}^n$*, and let* $\boldsymbol{W}\in\mathbb{R}^{n\times n}$ *be copositive. If*

$$\boldsymbol{w}^T\mathrm{LCP}(\boldsymbol{W},\boldsymbol{0})\geq 0\,,$$

then $\text{LCP}(\boldsymbol{W}, \boldsymbol{w})$ *contains a solution which Lemke's Algorithm can find in finite time.*

Finding a solution to such problems is in general not easy, and it is known that even with the structure that frictional contact affords, that solving LCP's is an NP-Hard problem (Baraff, 1991). While solution uniqueness is not guaranteed, if mapping the solution through a matrix $\boldsymbol{A}$ produces uniqueness, it also produces Lipschitz continuity:

Proposition 4 (Facchinei and Pang (2003))**.** *For all matrices* $\boldsymbol{W} \in \mathbb{R}^{n \times n}$*,* $\boldsymbol{A} \in \mathbb{R}^{m \times n}$*, if the function* $\boldsymbol{f}(\boldsymbol{w}) = \boldsymbol{A}\text{LCP}(\boldsymbol{W}, \boldsymbol{w})$ *is unique over a convex domain* $\Omega \subseteq \mathbb{R}^n$*, it is also Lipschitz on* Ω*.*

As mentioned in the previous subsection, LCP's are in general hard to solve, and even the copositive structure in frictional contact LCP's is not enough to avoid NP-Hardness Baraff (1991). However, some classes of LCP's are more tractible; namely, for positive semi-definite $\boldsymbol{W}$, any LCP $\text{LCP}(\boldsymbol{W}, \boldsymbol{w})$ can be either solved or certified to be unsolvable by solving a single convex QP in polynomial time Cottle et al. (2009). Pursuant to these properties, Anitescu (2005) developed an approximation to the LCP of Stewart and Trinkle (1996) which approximates normal complementary and maximal dissipation with the cone complementarity constraints

$$L_i \ni \boldsymbol{\Lambda}_i \perp \tilde{\boldsymbol{\phi}}_i(\boldsymbol{q}, \boldsymbol{v}') \in \text{dual}(L_i)\,, \tag{2.45}$$

$$\tilde{\boldsymbol{\phi}}_i(\boldsymbol{q}, \boldsymbol{v}') = \begin{bmatrix} \boldsymbol{\phi}_i(\boldsymbol{q}) \\ \boldsymbol{0} \end{bmatrix} + \boldsymbol{J}_i \boldsymbol{v}' \Delta t\,, \tag{2.46}$$

which can be captured and tractibly solved as a convex second-order cone program. This program can also be described in the context of optimization based dynamics, with the same $\boldsymbol{g}$ and $\mathcal{K}$ as Stewart and Trinkle (1996) and

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}', \boldsymbol{\Lambda}) = \sum_i \frac{1}{2}\text{Dist}(\tilde{\boldsymbol{\phi}}_i(\boldsymbol{q}, \boldsymbol{v}'), \text{dual}(L_i))^2 + \tilde{\boldsymbol{c}}_i(\boldsymbol{q}, \boldsymbol{v}') \cdot \boldsymbol{\Lambda}_i + \boldsymbol{d}_i(\boldsymbol{v}') \cdot \boldsymbol{\Lambda}_i\,, \tag{2.47a}$$

$$\boldsymbol{c}_i(\boldsymbol{q}, \boldsymbol{v}') = \begin{bmatrix} \max(\left\|\bar{\boldsymbol{\phi}}_i(\boldsymbol{q}, \boldsymbol{v}')\right\|, \left\|\boldsymbol{J}_{t,i}\boldsymbol{v}'\right\|_2 \Delta t) \\ \frac{1}{\boldsymbol{\mu}_i}\boldsymbol{J}_{t,i}\boldsymbol{v}'\Delta t \end{bmatrix}\,, \tag{2.47b}$$

where $\boldsymbol{d}_i$ is the same as in (2.40).

2.7. Empirical System Identification

System identification (sysID), or model learning, is the process by which a model's parameters $\boldsymbol{\theta}$ are fit to data $\mathcal{D} = \{(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}')_i\}_{i \in 1 \dots N}$. In general, a finite amount of *training* data $\mathcal{D}_{\text{train}}$ is available for this process. A ubiquitous method for recovering $\boldsymbol{\theta}$ from data is to empirically minimize the discrepancy, or *prediction error*, between the model's predictions and the training data:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}_{\text{train}}) \,, \tag{2.48}$$

$$\mathcal{L}_{\text{prediction}}(\boldsymbol{f}, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_i \left\| \boldsymbol{x}_i' - \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{u}_i) \right\|_2^2 \,. \tag{2.49}$$

(2.48) is often intractable to solve exactly, particularly when the dimension of $\boldsymbol{\theta}$ is large and when $\mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}_{\text{train}})$ is non-convex. In practice, this objective is only approximately minimized via some variant of stochastic gradient descent (SGD). In this algorithm, datapoints $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}')_i$ are iteratively chosen at random and $\boldsymbol{\theta}$ are updated via

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \left\| \boldsymbol{x}_i' - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{u}_i) \right\|_2^2 \,,$$

where $\alpha > 0$ is called the *learning rate* or *step size*. After this optimization process, the model (2.3) is embedded into robotics planning and control algorithms, in which the robot decides to encounter *test-time* trajectories $\mathcal{D}_{\text{test}}$. A critical assumption of these methods are that models that fit the training data well (e.g. have low prediction error) will also fit the test data similarly well. The discrepancy between the two is called the *generalization error*, equal to

$$\mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}_{\text{test}}) - \mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}_{\text{train}}) \,. \tag{2.50}$$

2.8. Data-Driven Modeling with Neural Networks

One popular design choice for learning robot dynamics is to model $\boldsymbol{f}_{\boldsymbol{\theta}}$ directly as a deep neural network (DNN). One simple type of DNN is a fully-connected feed-forward neural network, also

called a Multilayer Perceptron (MLP). Such networks map their inputs $\boldsymbol{r} \in \mathbb{R}^m$ to their outputs $\boldsymbol{y} \in \mathbb{R}^n$ via

$$\boldsymbol{z}_1 = \boldsymbol{\sigma}(\boldsymbol{W}_{\boldsymbol{r},1}\boldsymbol{r} + \boldsymbol{b}_1) \in \mathbb{R}^w \,, \tag{2.51a}$$

$$\boldsymbol{z}_k = \boldsymbol{\sigma}(\boldsymbol{W}_{\boldsymbol{z},k}\boldsymbol{z}_{k-1} + \boldsymbol{W}_{\boldsymbol{r},k}\boldsymbol{r} + \boldsymbol{b}_k) \in \mathbb{R}^w \,, \tag{2.51b}$$

$$\boldsymbol{y} = \boldsymbol{W}_{\boldsymbol{r},d}\boldsymbol{z}_{d-1} + \boldsymbol{W}_{\boldsymbol{r},d}\boldsymbol{r} + \boldsymbol{b}_d \,, \tag{2.51c}$$

$\boldsymbol{W}_{\boldsymbol{r},k}, \boldsymbol{W}_{\boldsymbol{z},k}$ are parameter matrices called weights; and $\boldsymbol{b}_i \in \mathbb{R}^w$ are parameter vectors called biases; and w and d are called the width and depth of the network. $\boldsymbol{\sigma}$ is an element-wise nonlinear function called an activation. Such DNN's are known to be universal function approximators, in that any function can be approximated to arbitrary precision by a DNN of sufficient size with a particular set of parameters. A a corresponding challenge in applying DNN's to system identification is that many distinct sets of parameters may fit the training data well, despite having very different predictions on the test data. DNN architectures and training schemes typically have an inductive bias by which a particular set of parameters is chosen. Common training techniques like stochastic gradient descent (SGD) as well as common regularizers such as weight-decay and spectral normalization typically have an inductive bias towards the smoothest function that fits the training data Belkin et al. (2019); Ribeiro et al. (2020); Zhang et al. (2019); Miyato et al. (2018).

2.9. Set Approximation via Sampling

Problems in robotics can often be approximately solved with arbitrary precision via stochastic sampling (e.g. planning with RRT* (Karaman and Frazzoli, 2011)). When the output to be captured is a set, precise approximation may be captured as an *ε-net*:

Definition 5. *For $\varepsilon \geq 0$, an ε-net of a set $\mathcal{X}$ is a set $\mathcal{X}' \subseteq \mathcal{X}$ such that for each $x \in \mathcal{X}$, $\exists x' \in \mathcal{X}'$ with $\|x - x'\|_2 \leq \varepsilon$.*

In the spirit of probabilistic completeness, in Chapter 7 we will compute sampling-based approximations of set-valued model predictions. We will prove this by leveraging a similar behavior for the image of a box under a Lipschitz continuous function:

Proposition 5 (Dense Sampling (Appendix E.2.1)). *Let $g(x) : \mathbb{R}^n \to \mathbb{R}^m$ be Lipschitz with constant L. Consider a set of N uniform i.i.d. samples $\mathcal{X} = \{x_1, \ldots, x_N\}$ from $[0, h]^n$. Then $g(\mathcal{X})$ is an ε-net of $g([0, h]^n)$ with probability at least*

$$1 - \frac{(1-\Omega)^N}{\Omega}, \qquad \Omega = \left\lceil \frac{hL\sqrt{n}}{\varepsilon} \right\rceil^{-n}. \tag{2.52}$$

CHAPTER 3

Related Work

3.1. Physics-based System Identification of Stiff Contact Dynamics

Modeling robots as rigid-body systems, and fitting parameters such models to data, has a long history in robotics; for an overview, see Khalil and Dombre (2002). Classical methods were primarily concerned with identifying scalar-valued inertial parameters such as link lengths; masses; and moments of inertia (Khosla and Kanade, 1985) of a robot making no contact with its environment, such as a robot arm moving in free space. Recent work has focused on both learning these parameters through contact interactions, as well as learning parameterizations of contact itself. As described in Section 2.7, such methods typically cast the sysID process as an optimization problem minimizing prediction error. When the model $\boldsymbol{f_\theta}$ is a physics simulator, this process is also referred to as *differentiable physics*. It has been long observed that minimizing prediction error through contact behaviors is numerically challenging (Kolev and Todorov, 2015; Fazeli et al., 2017b; Heiden et al., 2020; Le Lidec et al., 2021). Typically, methods overcome this challenge with advanced optimization techniques, such as second-order (Kolev and Todorov, 2015); constrained (Fazeli et al., 2017b); or global (Heiden et al., 2020; Antonova et al., 2023) methods. These methods are not tractable when applied to high-dimensional optimization. As a result, successful application of these methods have so far been limited to scenarios where the number of unknowns is small. By contrast to these methods, this work will use simple, tractable, first-order SGD-based optimization for the sysID problem, and instead overcome numerical challenges by replacing prediction error with an alternative objective that has a better-behaved optimization landscape.

3.2. Combining Physical Modeling with Neural Networks

As in this thesis, much recent work has concerned combining the structure of physics-based models and simulators with the flexibility of empirical deep learning. Such methods typically follow a *grey-box* modeling paradigm, in which part of the structure is learned, while another is derived from physics. Much of these methods are ultimately complementary to the work in this thesis, and we

highlight the potential relationships.

Some methods focus on structured learning of the continuous terms (i.e. $\boldsymbol{M}(\boldsymbol{q})^{-1}F_s(\boldsymbol{x},\boldsymbol{u})$) in the dynamics. Lutter et al. (2019) and Gupta et al. (2020) for instance induce the Lagrangian structure on neural network representations of $\boldsymbol{M}$ and F_s. This can manifest for instance by enforcing $\boldsymbol{M}$'s positive definite structure by learning its Cholesky decomposition, or by learning potential forces as the gradient of a potential energy neural network $U(\boldsymbol{q})$. This work by contrast focuses on novel representations for the contact terms, and such models could readily be used in conjunction with one another by combination in the manipulator equations (2.8).

Other work focuses not on learning the parametrization of an individual system, but on higher-accuracy or lower-computational-burden laws of motion to which known physical parameters are inputs. In particular a number of of works have built off of Interaction Networks (Battaglia et al., 2016), which encodes objects into Graph Neural Networks (GNN's) as nodes valued as their physical properties and states. Much recent work has built on this idea, in which high-accuracy, reusable, GNN-based physics laws for interacting rigid objects are developed. These methods rely on explicit parameterizations of object geometries as a group of particles (Li et al., 2019; Sanchez-Gonzalez et al., 2018, 2020) or meshes (Allen et al., 2022, 2023). The goal pursued in this work by contrast is partially to identify representations from object geometries, which might then be exported into practical or mesh format for use with these learned physical laws.

Other methods consider *residual learning* (Heiden et al., 2020; Ajay et al., 2018; Zeng et al., 2019; Fazeli et al., 2017c, 2020a) in which the error between a calibrated physical model and the real-world is learned as an unstructured model, such as a neural network or Gaussian process. The goal of this augmentation is to capture effects not captured in the physical model due to a lack of fidelity or precise identification. Our methods output such physical models that may serve as inputs to these methods.

3.3. Implicit Neural Representations of Object Geometries

By philosophical contrast to the methods described in the previous section, the culmination of our work is to approximate physically-grounded, infinite-dimenioal unknowns with neural networks, such as object geometries. Our work comes in context of much work in the computer vision community on implicit representation of geometries via neural networks. Deep Signed Distance Fields (SDF's) (Park et al., 2019) for instance learn the distance to an object in 3D space as a DNN. Neural Radiance Fields (NeRF's) (Mildenhall et al., 2020) alternatively directly learn the distribution of opacity and color properties of a scene. While they do not strictly coincide, opacity is correlated with the occupancy of space with material, and this correlation has been used to embed static ground-truth NeRFs into simulation (Cleac'h et al., 2023). Most recently, real-world quasi-static object interactions were learned by combining a GNN structure similar to interaction networks with learned NeRF representations of a small group of objects (Driess et al., 2023).

As we detail in Chapter 6, we choose a novel, convex support function-based representation which similar to these representations is best suited for its application. SDF's and and NeRF's by their structure are naturally suited for rendering, whereas our representation is designed for use in physical simulations, which themselves typically encode objects via convex decomposition Coumans (2015). Most similar to our representation is CvxNet (Deng et al., 2020), which also models convex decompositions via a DNN. CvxNet focuses on discovering a low-dimensional, latent shape space for a class of similar objects, such as chairs. Each object instance can be given to the network as a shape vector, an the output is an explicit, static number of hyperplane inequalities which parameterize the convex parts of the shape. By contrast, we focus on high-fidelity identification of individual shapes from motion capture data. We therefore instead use a novel DNN to flexibly capture each convex shape by directly modeling its support function.

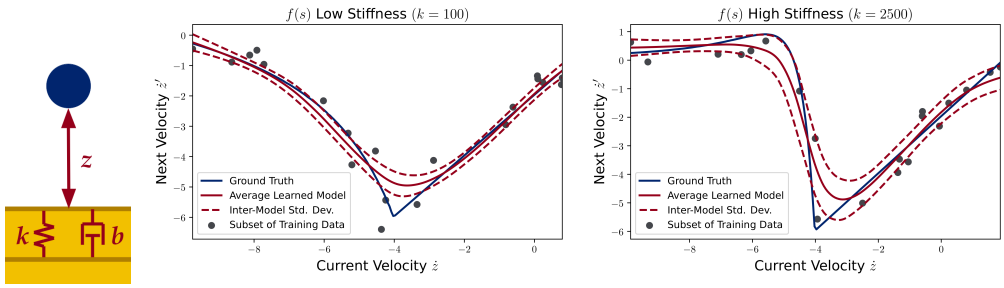
CHAPTER 4

Empirical Challenges in Deep Learning for Stiff Contact Dynamics

Parts of this chapter were previously published as parts of Mihir Parmar*, Mathew Halm*, and Michael Posa. Fundamental Challenges in Deep Learning for Stiff Contact Dynamics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, March 2021. URL https://ieeexplore.ieee.org/document/9636383, ©IEEE 2021. Mihir Parmar conducted the numerical experimentation and visualization, and wrote a large portion of Sections 4.3.1 and 4.3.2. The high-level experimental direction; conceptual and quantitative analysis; and remaining discussion were original contributions.

4.1. Introduction

As discussed in Chapter 1 and Sections 2.7 and 2.8, deep neural networks offer a simple and flexible way to synthesize an empirical model of a dynamical system. However, they present two main challenges when the dynamical system of interest is driven by stiff contact behavior. First, the model fitting process is typically conducted with stochastic gradient descent, a first-order optimization method not well-suited to fitting data from a stiff dynamical system. Second, DNN's typically extrapolate from training data to test data using a smoothness prior, in direct conflict with the nearly-discontinuous nature of the true dynamics. This smoothing effect is particularly harmful when available data is sparse, as can be see in a 1D example in Figure 4.1. If stiffness indeed significantly influences the performance of deep learning, then serious questions must be raised not only about robotic learning methods, but also the relevance of simulated results to the robotics community. Many simulators, such as MuJoCo (Todorov, 2014), allow users to specify mechanical stiffness; properly used, simulation can produce physically-accurate behaviors Kolev and Todorov (2015); Erez et al. (2015). However, ubiquitous benchmarking suites often use software default values for stiffness, rather than values tuned for realism Brockman et al. (2016). The idea that unrealistic contact settings can generate a gap between simulated and real-world performance is an existing intuition in the robotics community (Tedrake).



(a) (b)

Figure 4.1: Challenges of learning stiff dynamics are shown on a 1-D example. (a): A point mass (blue) falls from an initial height $z_t = 1$ toward compliant ground (yellow), modeled as a spring-damper system. (b): For each of two stiffnesses (k), 100 predictive models are trained on noisy data to predict the next velocity $\dot{z}_{t+1}$ from different initial velocities $\dot{z}_t$. Learning performance is heavily degraded on the stiffer $k = 2500$ system, despite meticulous hyperparameter optimization; training loss, ground-truth prediction error, and inter-model variance are 197%, 413%, and 309% higher than for $k = 100$. Details of this experiment may be found in Appendix B.1.

In this chapter, we contribute an empirical quantification and isolation of the detrimental effects of stiffness on deep learning performance. We begin by describing how stiffness enters into the equations of motion of a simple simulated system in MuJoCo in Section 4.2, and show that default settings are significantly less stiff than many real systems. In Section 4.3, we propose a testing methodology to examine the negative effects of stiffness on inherent unpredictability, training process degradation, generalization, and long-term prediction. As is common intuition, our results (Section 4.4) show that raising stiffness degrades ground-truth model predictions as the underlying system becomes more sensitive to noise. However, we find that stiffness induces multiple pathological behaviors beyond this effect:

1. The training error of learned models degrades with stiffness nearly twice as fast as the ground-truth model, even for single-step predictions.

2. While generalization error can be eliminated for non-stiff systems with ample training data, test error stays significantly higher than training error for stiff systems.

3. Data-efficiency degrades 100-fold for our stiffest models when evaluated on long-term prediction.

These results raise serious questions about whether existing methods effectively utilize deep learning's most powerful and essential behaviors, and whether typical simulated environments faithfully capture essential challenges of real-world physics. In Section 4.6, we suggest opportunities for further experimentation in this area.

4.2. Example System

We now describe a simple example system and associated data generation methodology, for which we will isolate the effects of stiffness on learning performance.

4.2.1. Simulation Environment

While the many uncontrollable factors of real-world experiments offer a challenging environment to test newly-developed algorithms' performance, the primary goal of this chapter is instead to isolate the effects of stiffness on commonplace methods in robotics. Unmodeled material complexities and unknowable noise distributions in a real robotic system would therefore befuddle the results presented here, rather than strengthen them. We therefore conduct our experiments in a simulated environment, which allows them to be easily repeated or used as a benchmarking task in future research. We conduct our experiments MuJoCo (Todorov, 2014), because it allows for direct control over contact stiffness. Using MuJoCo also enhances the relevance of our results to ubiquitous benchmarking suites which use the simulator (Brockman et al., 2016; Tassa et al., 2020).

While MuJoCo models objects as being exactly rigid, it allows for an interpretation of mechanical stiffness by using a "soft contact" model similar to the one used in Figure 4.1; a detailed discussion can be found in Todorov (2014) and in the online MuJoCo documentation[1]. MuJoCo solves for appropriate contact forces with a convex optimization problem, and thus there is no closed form expression for the forces as a function of the current state. However, when an object makes contact with a static environment, inter-body penetration $\boldsymbol{\phi}$ approximately[2] obeys

$$\ddot{\boldsymbol{\phi}} \approx -b\frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} - k\boldsymbol{\phi}\,. \tag{4.1}$$

[1] http://www.mujoco.org/book/computation.html

[2] A more detailed treatment of this behavior can be found in Appendix B.2.

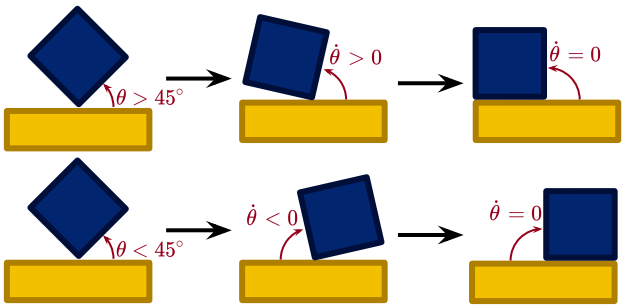


Figure 4.2: Sensitivity to initial conditions and near-instantaneous impact of a 2D block on flat ground are shown. (left) Two trajectories begin from nearly identical initial conditions, where the block (blue) contacts the ground (yellow) at 1 corner; the center of mass is left of the contact point in the upper trajectory and to the right in the lower one. (center) after some time has elapsed, the state of the cube differs drastically in the two trajectories. (right) the velocity of the cube jumps to zero abruptly as it hits the ground in both cases.

Here, the "stiffness" k is the primary mechanism resisting penetration, and the damping ratio $\zeta = \frac{b}{2\sqrt{k}}$ controls elasticity of impacts. Similar techniques have long been used for stable simulation of constrained dynamical systems, dating back to Baumgarte (1972). We note that the units of k are $\frac{\text{N}}{\text{kgm}}$, whereas mechanical stiffness is typically expressed in $\frac{\text{N}}{\text{m}}$ units. MuJoCo's default values for k are in the 2000–2500$\frac{\text{N}}{\text{kgm}}$ range. However, as we will discuss in Section 4.2.3, the corresponding contact behavior is far softer than that of many real-world objects, including common robotic platforms.

4.2.2. System Description

High-dimensional systems, much like real-world environments, are also commonly used to stress-test new robot learning algorithms (Brockman et al., 2016). By contrast, in this chapter, picking a simple, low-dimensional system instead allows us to more thoroughly and tractably analyze the effects of stiffness under reduced computational and sample complexity. We follow previous studies (Jiang et al., 2018; Fazeli et al., 2020a), and choose a "die roll" system, in which a single, rigid cube makes contact with the ground. Despite being low-dimensional, the cube exhibits many of the hallmark challenges in contact modeling: stick-slip transition, discontinuous impact, multiple contact points, and extreme sensitivity to initial conditions; Figure 4.2 illustrates some of these behaviors in 2D.

Our 3D die system has a 13-dimensional state

$$\boldsymbol{x} = \Big[\boldsymbol{p}\,;\quad \boldsymbol{q}\,;\quad \dot{\boldsymbol{p}}\,;\quad \boldsymbol{\omega}\Big]\,, \tag{4.2}$$

where $\boldsymbol{p} \in \mathbb{R}^3$ is the center of mass position; $\boldsymbol{q} \in S^3$ is the orientation of the cube, expressed as a quaternion; The generalized velocity $\boldsymbol{v} = [\dot{\boldsymbol{p}}\,;\boldsymbol{\omega}]$ contains the world-frame c.o.m. velocity $\dot{\boldsymbol{p}} \in \mathbb{R}^3$ and body-frame angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$. Similar to the construction in Section 2.5.4, MuJoCo approximates the manipulator equations (2.8) in discrete-time with a semi-implicit formulation Todorov (2014). As this system has no inputs, its dynamics have the form

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t)\,, \tag{4.3}$$

where t denotes the time index. For a symmetric cube[3], $\boldsymbol{f}$ has the simplified form

$$m(\dot{\boldsymbol{p}}_{t+1} - \dot{\boldsymbol{p}}_t) = (F(\boldsymbol{x}_t) - m\boldsymbol{g})\Delta t\,, \tag{4.4}$$

$$I(\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t) = \boldsymbol{T}(x_t)\Delta t\,, \tag{4.5}$$

$$\boldsymbol{p}_{t+1} - \boldsymbol{p}_t = \dot{\boldsymbol{p}}_{t+1}\Delta t\,, \tag{4.6}$$

$$\boldsymbol{q}_t^{-1} \otimes \boldsymbol{q}_{t+1} = \boldsymbol{Q}(\boldsymbol{\omega}_{t+1}\Delta t)\,, \tag{4.7}$$

where Δt is the time-step duration; m and I are the cube's mass and inertia; $\boldsymbol{g}$ is the gravitational acceleration vector; and F and $\boldsymbol{T}$ are the average contact force and torque over the time step. $\boldsymbol{Q}(\boldsymbol{r}) = [\cos\frac{\|\boldsymbol{r}\|_2}{2}; \hat{\boldsymbol{r}}\sin\frac{\|\boldsymbol{r}\|_2}{2}]$ is the quaternion corresponding to a rotation of angle $\|\boldsymbol{r}\|_2$ about axis $\hat{\boldsymbol{r}} = \frac{\boldsymbol{r}}{\|\boldsymbol{r}\|_2}$ and $\otimes$ is the quaternion product. We use system parameters that closely match the real system used in (Pfrommer* et al., 2020); a full list can be found in Table 4.1.

4.2.3. Data Generation

In order to isolate how different stiffnesses k generate different behaviors, for each of 3 stiffnesses listed in Table 4.2, we generate a dataset $\{\boldsymbol{\tau}\}$ of "dice roll" trajectories $\boldsymbol{\tau} = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, ...\boldsymbol{x}_{t_f}\}$ with

[3]the symmetry assumption implies that the Coriolis forces are zero, as the inertia tensor is a multiple of the identity matrix.

Table 4.1: Simulated Die Roll Parameters for Empirical Study

Constant	Symbol	Value	Units
mass	m	0.37	kg
inertia	I	6.167×10^{-4}	$\mathrm{kg\,m^2}$
side length	l	0.1	m
gravity	g	9.81	$\frac{\mathrm{m}}{\mathrm{s}^2}$
friction coefficient	μ	1	(none)
stiffness	k	(varies)	$\frac{\mathrm{N}}{\mathrm{kgm}}$
damping ratio	ζ	1.04	(none)
time-step	Δt	6.74×10^{-3}	s

an identical process, summarized here and detailed further in Appendix B.2. We refer to the three stiffnesses as *Hard*, *Medium* and *Soft*.

We instantiate the system for a given stiffness in MuJoCo with the parameters in Table 4.1; the damping coefficient b is selected to keep the damping ratio ζ consistent between stiffnesses. Initial states are sampled uniformly around a nominal state $\boldsymbol{x}_{0,ref}$. From the initial state, we simulate forward in time with MuJoCo's dynamics (4.4)–(4.7) until the cube impacts the ground and comes to rest.

In the real world, position and orientation of similar systems are commonly tracked via computer vision, which can incur a small amount of slowly-drifting measurement noise (Pfrommer* et al., 2020). To approximate this error, for each trajectory, we add a small, uniformly random offset to the entire trajectory, and a second round of smaller noise independently to each datapoint. Finally, velocity states are reconstructed using the finite difference equations (4.6)–(4.7) on the noisy configurations. The total noise injected through this process on position, orientation, linear velocity, and angular velocity is on the order of 1 mm, deg, $\frac{\mathrm{mm}}{\mathrm{s}}$, and $\frac{\mathrm{deg}}{\mathrm{s}}$, respectively.

For each stiffness setting, we collect $10,000$ trajectories $\{\tau\}_{\text{train}}$ for hyperparameter optimization and training purposes, and $1,000$ more trajectories $\{\tau\}_{\text{eval}}$ for evaluation of the optimized models. To evaluate physical realism of each of these settings, we also compute the maximum ground penetration of the die, averaged over trajectories (Table 4.2 and Figure 4.3). Even for the *Hard* stiffness, which is comparable to MuJoCo's default, we observe ground penetration of around 10% of the die body-

Table 4.2: Stiffnesses and Ground Penetration for Simulated Die Rolls

Stiffness Setting	k $\left(\frac{\text{N}}{\text{kgm}}\right)$	Max Penetration (mm)
Hard	2500	12
Medium	300	26
Soft	100	40

length. By comparison, deformations on real-world objects are often imperceptible to the human eye. Thus, even our *Hard* model is far less stiff than the real-world dynamics upon which our system was based (Pfrommer* et al., 2020).

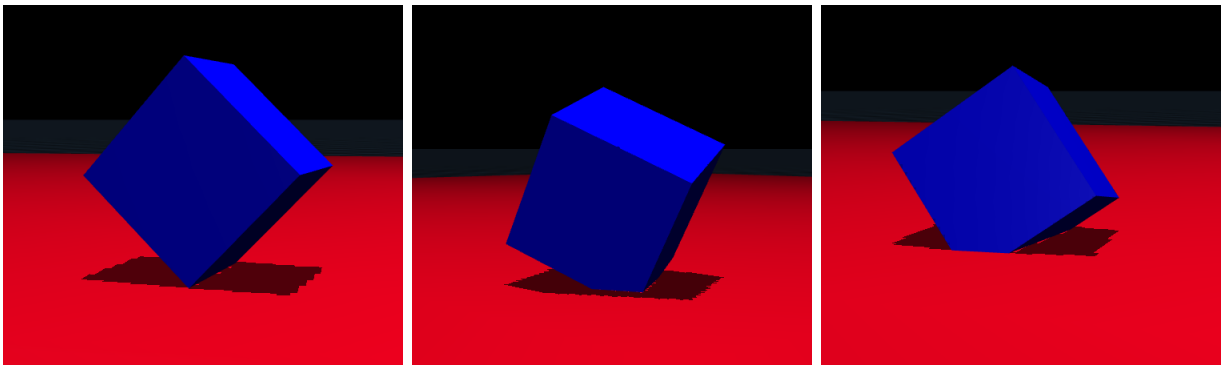


Figure 4.3: From left to right, illustration from MuJoCo demonstrating the variation in the amount of ground penetration for *Hard*, *Medium*, and *Soft* settings, respectively. These visualizations are captured from trajectories with identical initial states.

4.3. Experimental Procedure

We now describe in detail the process of learning a dynamical system from data; hypothesis about the specific effects that stiffness has on dynamics learning; and motivate the design of our experiments. The Pytorch codebase is accompanied by further details and experiments online[4].

4.3.1. Representing Second-order Systems with Recurrent Neural Networks

We consider the approach previously discussed in Chapter 1 and Sections 2.7 and 2.8, in which the entire dynamics update function is learned as a DNN.

We consider two common neural network types applied to this domain. The first and most elementary, as discussed in Section 2.8, is to map $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t+1}$ with a simple multilayer perceptron $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_t)$

[4]https://sites.google.com/view/contact-learning-bias

(Janner et al. (2019); Nagabandi et al. (2019)). However, this method has been shown to struggle with noisy data in a manipulation setting Ajay et al. (2019); Ajay et al. (2018), an expected behavior given the sensitivity of contact dynamics Kolev and Todorov (2015). An intuition in robotics is that better estimates of the current state can be generated by fusing multiple sensor readings. Correspondingly, a common approach is to use recurrent neural networks (RNN's) $\boldsymbol{f_\theta}\left(\boldsymbol{x}_{t-h+1:t}\right)$, which instead take as input a time window $\boldsymbol{x}_{t-h+1:t} = (\boldsymbol{x}_{t-h+1}, \ldots \boldsymbol{x}_t)$ with history length $h > 1$ Ajay et al. (2019); Ajay et al. (2018), We will also therefore consider three common RNN variants as well: Long Short Term Memory (*LSTM*) (Hochreiter and Schmidhuber, 1997), Gated Recurrent Units (*GRU*) (Cho et al., 2014) and Bi-directional LSTM (*BiLSTM*) (Graves et al., 2005).

Much like the method described in Section 2.5.4, MuJoCo predicts the next velocity $\boldsymbol{v}_{t+1}$, and then constructs the next configuration using the finite-differencing method given by (4.6)–(4.7) Todorov (2014). Since our velocity data is generated with the same finite-difference, it is sufficient to output either $\boldsymbol{f_\theta}\left(\boldsymbol{x}_{t-h+1:t}\right) \approx \boldsymbol{v}_{t+1}$ or $\boldsymbol{f_\theta}\left(\boldsymbol{x}_{t-h+1:t}\right) \approx \Delta\boldsymbol{v} = \boldsymbol{v}_{t+1} - \boldsymbol{v}_t$ from the network, and then reconstruct the next configuration with (4.6)-(4.7) as MuJoCo does. Via extensive hyperparameter optimization, the optimal network structure and target variable choice for all stiffness setting were determined to be GRU and $\boldsymbol{v}_{t+1}$ respectively, as listed in Table 4.3. A detailed explanation of the process is given in Appendix B.3.

4.3.2. Training Process

To train one of our networks, we first aggregate a set of N trajectories $\{\boldsymbol{\tau}_{1:N}\}$ randomly sampled from $\{\boldsymbol{\tau}\}_{\text{train}}$ and slice them into training data inputs $\{\boldsymbol{x}_{t-h+1:t}\}$ and corresponding outputs $\{\boldsymbol{v}_{t+1}\}$. To improve numerical conditioning during training, we follow a standard procedure of normalizing the input data to have zero mean and unit variance Nagabandi et al. (2019). We further split the sliced data $\{(x_{t-h+1:t}, v_{t+1})_i\}$ in 70:20:10 proportions into training ($\mathcal{D}_{\text{train}}$), validation ($\mathcal{D}_{\text{valid}}$) and test ($\mathcal{D}_{\text{test}}$) sets.

For a dataset $\mathcal{D}$, as our models only output velocity variables, it is sufficient to consider the prediction

Table 4.3: Optimized Hyperparameters for Learning Simulated Cube Dynamics

Hyperparameter	Stiffness Setting		
	Hard	*Medium*	*Soft*
Network architecture	GRU	GRU	GRU
Target variable	v_{t+1}	v_{t+1}	v_{t+1}
learning-rate	1e-4	1e-5	1e-5
hidden-size	128	128	128
history-length	16	16	16
weight-decay	0	4e-5	4e-5

loss over $\mathcal{D}$ for a model $\boldsymbol{f}$ as

$$\mathcal{L}_{\text{prediction}}(\boldsymbol{f}, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_i \|(\boldsymbol{v}_{t+1})_i - \boldsymbol{f}\left((\boldsymbol{x}_{t-h+1:t})_i\right)\|_2^2 \tag{4.8}$$

Accordingly, we train our dynamics models using the Adam optimizer (Kingma and Ba, 2015), a state-of-the-art extension of SGD, to minimize $\mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}_{\text{train}})$. We terminate training with early stopping with a patience of 30 epochs, and save the model with the lowest validation loss $\mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}_{\text{valid}})$. Test set error $\mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}_{\text{test}})$ is then used as the metric during hyperparameter optimization. To provide an optimistic perspective on how the dynamics of each stiffness setting can be learned, we optimize separate hyperparameters for each stiffness. This process is detailed in Appendix B.3, and Table 4.3 specifies the final set of selected hyperparameter values for each of the stiffness setting.

4.3.3. Measuring Stiffness's Effect on Learning Performance

To perform an optimistic analysis on how well learning algorithms perform on systems with different stiffnesses, we focus performance analysis on our hyperparameter-optimized models in three settings: the effectiveness of Adam in minimizing the training set loss; generalization performance; and long-term prediction performance.

Broadly, the goal of supervised learning methods, including our dynamics learning process, is to generate accurate outputs for unseen inputs. In dynamics learning, this behavior is typically quantified as low test error $\mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}_{\text{test}})$. Our discussion in Chapter 1 suggests that the test error

will increase for higher stiffness settings, but it is important to note that the test set error can be driven up by many mechanisms. Inspired by the separate treatment of approximation, estimation, and generalization error common in statistical learning theory (Poggio and Smale, 2003), we now define, motivate, and hypothesize about the following decomposition of the test error:

$$\begin{aligned}\mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}_{\text{test}}) = \mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\text{oracle}}, \mathcal{D}_{\text{train}}) \\ + (\mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}_{\text{train}}) - \mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\text{oracle}}, \mathcal{D}_{\text{train}})) \\ + (\mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}_{\text{test}}) - \mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}_{\text{train}}))\ .\end{aligned} \tag{4.9}$$

As discussed in Section 4.1, it is important to acknowledge that even if the *exact* MuJoCo model used to generate the data is simulated from a noisy initial condition, it will make an imperfect prediction of the future states. Furthermore, it is well known that stiff dynamics will exacerbate this issue (Chatterjee, 1997; Kolev and Todorov, 2015; Pfrommer* et al., 2020). We capture this effect with the first term in the test error decomposition (4.9), $\mathcal{L}_{oracle} = \mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\text{oracle}}, \mathcal{D}_{\text{train}})$. To be precise, our *MuJoCo oracle* model $\boldsymbol{f}_{\text{oracle}}(\boldsymbol{x}_{t-h+1:t})$ predicts the next velocity $\boldsymbol{v}_{t+1}$ with MuJoCo using the current noisy state $\boldsymbol{x}_t$ and the Netwton-Euler equations (4.4)-(4.5). As the *MuJoCo oracle* captures the underlying true behavior of the system, it serves as a natural, optimal baseline for the learned models.

Since prediction loss can be poorly conditioned when learning stiff dynamics (Heiden et al., 2020; Pfrommer* et al., 2020), we hypothesize that Adam will have difficulty in converging to good minima consistently; therefore, the training loss at convergence of stiffer models is likely to have higher mean. To test this hypothesis, we train models on data with different stiffnesses and dataset sizes, and then observe how much the resulting training error is degraded in comparison to the performance of the MuJoCo oracle: $(\mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}_{\text{train}}) - \mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\text{oracle}}, \mathcal{D}_{\text{train}}))$.

Deep learning is biased towards fitting a smooth interpolator on the data Belkin et al. (2019); Ribeiro et al. (2020); however, as we noted in Chapter 1, the underlying behavior of contact is non-smooth. Hence, at equal training set sizes, we expect that DNNs fit to stiffer systems' data will

suffer worse generalization error. We examine this hypothesis by comparing the generalization error ($\mathcal{L}_{\text{prediction}}(\boldsymbol{f_\theta}, \mathcal{D}_{\text{test}}) - \mathcal{L}_{\text{prediction}}(\boldsymbol{f_\theta}, \mathcal{D}_{\text{train}})$) of learned models corresponding to different stiffnesses and dataset sizes.

While we have followed a commonly used approach by training our models on single-step predictions (e.g. Chua et al. (2018); Nagabandi et al. (2019); Tedrake), long-term prediction quality is essential for model-based control methods, such as MPC Chua et al. (2018). We therefore additionally evaluate our models' long-term prediction capability. For a particular ground-truth trajectory $\boldsymbol{\tau} \in \{\boldsymbol{\tau}\}_{\text{eval}}$, we use the initial ($h$) ground-truth states $\{\boldsymbol{x}_t\}_{t=0}^{t=h-1} \in \boldsymbol{\tau}$ as input for the learned model and recursively construct predicted trajectory for next $\hat{t}_f$ time-steps $\{\hat{x}_t\}_{t=h}^{t=h+\hat{t}_f-1}$. We report the temporally-averaged absolute position and rotation error for each model:

$$e_{pos} = \frac{1}{\hat{t}_f} \sum_{j=h}^{h+\hat{t}_f-1} \left\| \hat{\boldsymbol{p}}_j - \boldsymbol{p}_j \right\|_2 , \qquad e_{rot} = \frac{1}{\hat{t}_f} \sum_{j=h}^{h+\hat{t}_f-1} \left| \text{angle}\left(\hat{\boldsymbol{q}}_j, \boldsymbol{q}_j\right) \right| ,$$

where, angle$(\cdot, \cdot)$ represents the relative angle between two quaternions.

To make a fair evaluation for models with different history length and simultaneously ensure that the prediction horizon is long enough to capture ground impact and block tumbling, we use $\hat{t}_f = 50$ (a 337 ms duration) in our experiments.

4.4. Results

In Table 4.4 we report *MuJoCo oracle*'s performance on the single-step velocity prediction $\mathcal{L}$oracle task across different stiffnesses. As expected, we observe that the single-step prediction performance of the *MuJoCo oracle* improves as the contact is made softer.

Table 4.4: *MuJoCo oracle* Prediction Performance

Stiffness Setting	$\mathcal{L}$oracle	e_{pos} (% width)	e_{rot} (deg)
Hard	$0.0836 \pm 1.8 \times 10^{-3}$	4.31 ± 0.13	3.98 ± 0.03
Medium	$0.011 \pm 1.8 \times 10^{-4}$	3.57 ± 0.08	3.26 ± 0.03
Soft	$3.19 \times 10^{-3} \pm 1.9 \times 10^{-5}$	2.92 ± 0.04	2.77 ± 0.03

While the single-step oracle error for *Hard* contact is nearly 20x higher when compared to *Soft*

contact, we find that it only accounts for approximately half of the training error. Fig. 4.4a demonstrates the difference in the converged training loss of learned models and the oracle single-step errors ($\mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}_{\text{train}}) - \mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\text{oracle}}, \mathcal{D}_{\text{train}})$) across different dataset sizes and different contact settings. The resulting values were right skewed and non-negative; therefore, we assume their distribution to be log-normal and construct its 95% confidence interval using Cox's method (Land, 1972). Stiffer models show worse average training loss across all tested dataset sizes with high confidence, with nearly 10x gap between the *Hard* and *Soft* models in the high data regime. Furthermore, there is a large data efficiency gap; *Hard* models trained on 5000 trajectories perform worse than both *Medium* and *Soft* models trained at just 100 trajectories.

In Fig. 4.4b, we plot the generalization error of the learned models with their 95% log-normal confidence intervals. Similar to the trends noted in Fig. 4.4a, we observe that the generalization error also exhibits a 10x gap in the high data regime, and that again *Hard* models perform worse than their *Medium* and *Soft* counterparts do with 50x less data. Additionally, while *Medium* and *Soft* models improve generalization by over a factor of 100 as the dataset size increased from 50 to 5000 trajectories, *Hard* models by contrast improve by less than a factor of 10.

We capture the long-term prediction errors (e_{pos}, e_{rot}) of the *MuJoCo oracle* in Table 4.4 and that of the learned models in Fig. 4.4c–4.4d. The MuJoCo oracle errors are at least 10x smaller than the learned models for both metrics. We also observe that for both *MuJoCo oracle* and the learned models, the errors increase for stiffer models. In the case of learned models, the *Hard* models perform worse when trained on up to 5000 trajectories than the *Soft* models perform only for 50, a data-efficiency gap of at least 100x.

4.5. Discussion

Our results provide compelling evidence that deep learning methods are negatively impacted by stiffness induced by contact. Of particular note is that this effect is significant when compared to the inherent uncertainty of predicting stiff dynamics for noisy data; training set error grows nearly *twice* as fast with stiffness as the *MuJoCo oracle* model (Fig. 4.4a). We also see in Fig. 4.4b that stiff dynamics can violate the common intuition that generalization error vanishes as the training

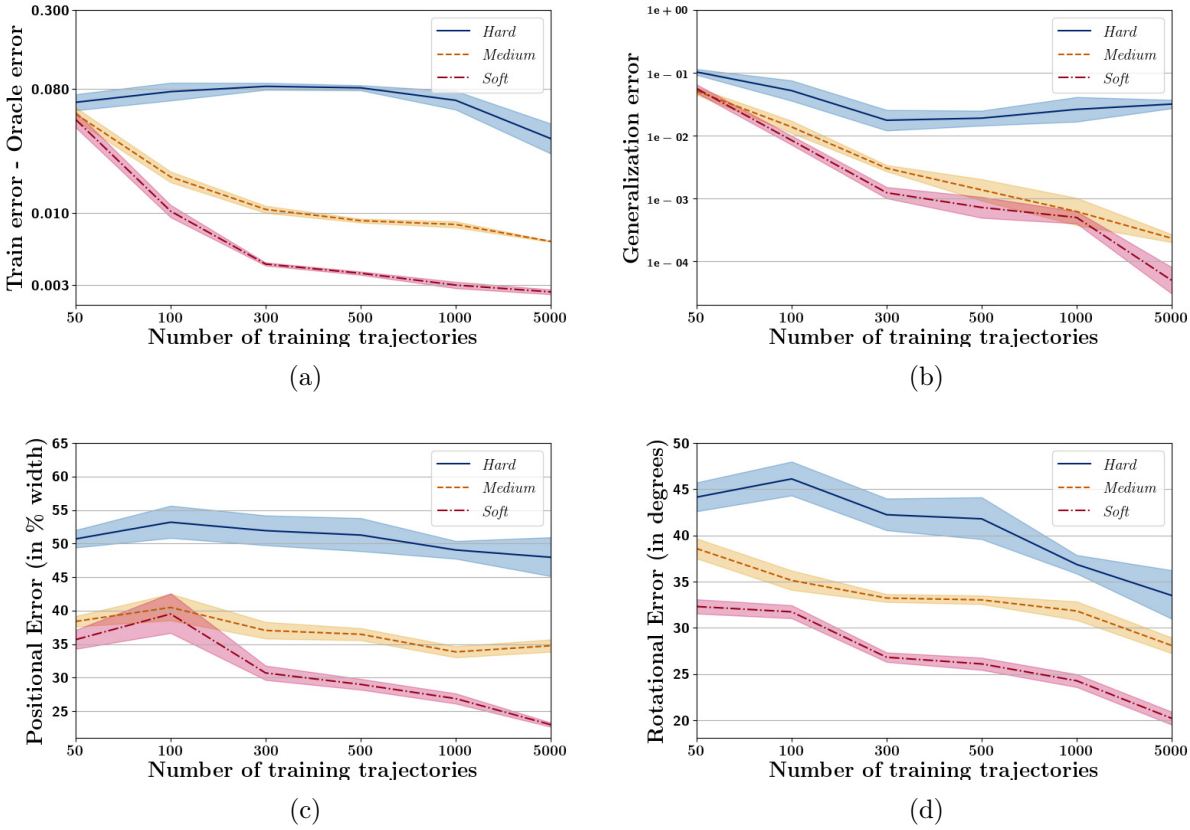


Figure 4.4: We compare the quality of learned models across stiffness settings and dataset sizes. We plot several metrics with 95% log-normal confidence intervals. (a): As stiffness increases, the gap between the training error and the oracle's error grows. (b): Stiffer models show a significantly higher average generalization error across dataset sizes. (c,d): We compare the performance of our optimized networks on long-term prediction of position and orientation. Data-efficiency of learning *Hard* models is at least 100x worse than for *Soft* models.

set size approaches infinity. While our softer models clearly behave as such, a 100x increase in training set size made little impact on generalization for *Hard* models.

It is also vital to understand how learning performance affects the downstream robotics task. Often, a goal in robotics is to gain an *accurate enough* model as fast as possible to use for long-term prediction. The bottleneck in this scenario is often training data collection (Nagabandi et al., 2019). From this data-efficiency perspective, increased stiffness has degraded learning performance 100-fold, as that the *Hard* models are still worse than the *Soft* models with 100 times more data (Fig. 4.4c–4.4d).

It is worth noting that long-term prediction performance can decorrelate from short-term predictions, especially for stiff dynamics Ajay et al. (2019); Halm and Posa (2019). Training directly on multi-step prediction error is therefore desirable. However, the stiffness of the training optimization problem grows *exponentially* in the prediction horizon Ribeiro et al. (2019), and we have seen here that even a prediction horizon of 1 is challenging on stiff systems.

4.6. Conclusion and Future Work

In this chapter, we have outlined a fundamental conflict between the dynamics of contact and common deep learning approaches, which significantly degrade the performance of learned models. While compelling, these results are only an initial study. Notably, our system has no actuation, and has lower dimensional state than many robotics tasks; the performance gap between hard and soft contact could be even wider for more complex systems. Future studies into more complex systems are vital to understanding this relationship. Additionally, as MuJoCo's compliant contact is only a coarse approximation of real-world deformation dynamics, real-world experiments which compare the dynamics of objects with different mechanical stiffness would strengthen the relevance of these results to real robotic systems. Data efficiency has also been a primary focus of several recent contributions to robotic learning. Some approaches have focused on training on long-term prediction Ajay et al. (2018); Ajay et al. (2019); and locally-accurate, task-specific models Chua et al. (2018). While none of these methods attempt to handle the conflict between stiffness and deep learning directly, examining effects of stiffness on these algorithms would strengthen the relevance of our results to the state-of-the-art.

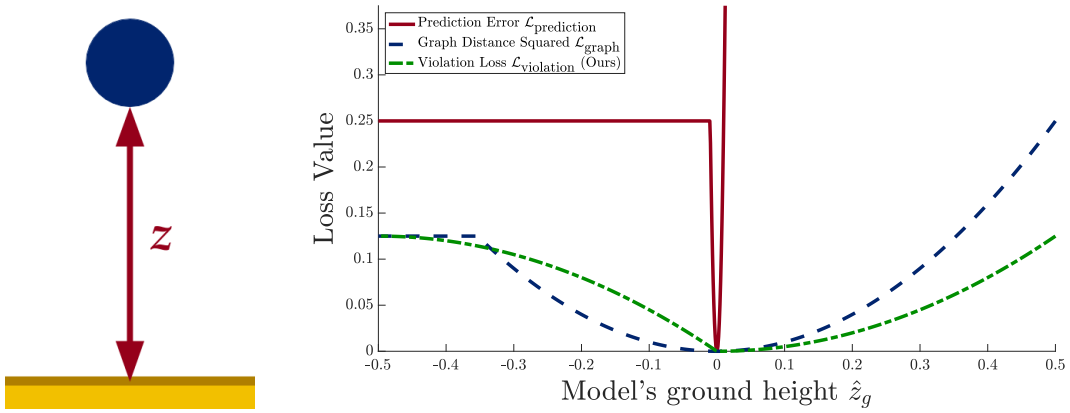
CHAPTER 5

A Mechanics-inspired, Non-stiff Loss for Learning Stiff Contact Dynamics

Parts of this chapter were previously published as parts of Bibit Bianchini, Mathew Halm, Nikolai Matni, and Michael Posa. Generalization bounded implicit learning of nearly discontinuous functions. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference (L4DC)*, volume 168 of *Proceedings of Machine Learning Research*, pages 1112–1124. PMLR, 23–24 Jun 2022. URL https://proceedings.mlr.press/v168/bianchini22a.html. Bibit Bianchini conducted a previous version of the numerical analysis on the 1-D example explored in Section 5.2. The remaining included portions represent original contributions.

5.1. Introduction

In Chapter 4, we have observed that prediction error induces poor numerical conditioning into the training process of data-driven models. As a result, models optimized with stochastic gradient descent may become stuck in local minima, and fail to even fit training data well. In this chapter, we raise questions about whether prediction error is even a high-quality metric of model quality for stiff systems. We begin by demonstrating pathologic behaviors of prediction error gradients on a 1D example (Section 5.2). We then provide a computational and statistical perspective on the shortcomings of prediction error as a suitable metric for stiff, optimization-based dynamics (Section 5.3). Inspired by recent work in control and planning through contact, we then propose a novel, mechanics-inspired alternative prediction error, and prove a statistical connection to classical errors-in-variables modeling (Cifarelli, 1988; Adcock, 1878) (Section 5.4), original derived in Pfrommer*, Halm*, and Posa (2020) and Bianchini, Halm, Matni, and Posa (2022). We demonstrate the superior optimization characteristics on a rigid, simulated example, analogous to the die roll system in Chapter 4 (Section 5.5). We conclude with some directions for future work in Section 5.6.



(a) (b)

Figure 5.1: Illustration of various model-fitting objectives on a 1D example. (a): A point mass (blue) falls from an initial height z and velocity towards rigid, flat ground at height $z_g = 0$. The mass makes an impact with the ground before the next time-step. (b): For the problem of estimating the ground height as a learnable parameter $\hat{z}_g$ from this single state transition, we display the optimization landscape the behaviors of 3 candidate losses, each of which equal 0 if and only if $\hat{z}_g = z_g$. Almost the entire landscape of prediction error (red) is either completely flat (zero gradient) or extremely stiff (near-infinite gradient), where as graph distance and our proposed loss are comparably well-conditioned.

5.2. Gradient Pathologies on a One Dimensional Example

To illustrate the difficulty of optimizing prediction error for rigid-body systems with contact, we consider the one-dimensional rigid contact system in Figure 5.1a. This system consists of a point unit mass at height z above flat ground at heigh $z_g = 0$, and has state $[z; \dot{z}]$. As there is no lateral movement, we may forgo analysis of friction and simplify the dynamics (2.35) to

$$\text{find} \qquad \dot{z}'; \mathbf{\Lambda}_n \,, \tag{5.1a}$$

$$\text{s.t.} \qquad \dot{z}' = \dot{z} - g\Delta t + \mathbf{\Lambda}_n \,, \tag{5.1b}$$

$$\mathbf{0} \leq \mathbf{\Lambda}_n \perp (z + \dot{z}'\Delta t) - z_g \geq \mathbf{0} \,. \tag{5.1c}$$

These dynamics are have the unique analytical solution

$$\boldsymbol{f}_{\boldsymbol{\theta}}(z, \dot{z}) = \dot{z}' = \dot{z} - g\Delta t + \text{pos}\left(-\dot{z} + g\Delta t + \frac{z_g - z}{\Delta t}\right) . \tag{5.2}$$

On inspection, we can see that these dynamics are stiff for small Δt, specifically in the sense that the Lipschitz constant both with respect to initial condition z and parameter z_g is $\frac{1}{\Delta t}$. Consider a single initial state with sufficient downward velocity to cause an impact: $[z; \dot{z}] = \left(\frac{\Delta t}{2}, g\Delta t - 1\right)$. Solving (5.2) results in $\dot{z}' = -\frac{1}{2}$. We consider the 1-D identification problem of finding an estimate $\hat{z}_g$ of z_g from this single datapoint, where Δt is set to 20[ms]. The standard method as described in Section 2.7 would be to do so with gradient descent w.r.t. $\boldsymbol{\theta} = \hat{z}_g$ on the prediction error $\mathcal{L}_{\text{prediction}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \{[z; \dot{z}], \dot{z}'\})$. The 1-D setting allows to plot the loss landscape for visualization, shown in red in Figure 5.1b. Here, we highlight two key challenges in this optimization process:

1. **Zero gradients**: When the modeled objects are much farther apart than the ground-truth geometries, predicting forward in time from even a noiseless initial condition may fail to observe that the objects can affect each others' motion. This corresponds to the left region of Figure 5.1b, where the predicted next state, and therefore the prediction error, is invariant to the geometric parameters. Examining (5.2), this is because, $\text{pos}\left(\dot{z} + g\Delta t + \frac{z_g - z}{\Delta t}\right)$ has zero gradient in this region. Locally, $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{prediction}} = \mathbf{0}$. As a result, large regions in parameter space where SGD may become stuck (i.e. local minima) may exist.

2. **Near-infinite gradients**: When the modeled objects are much closer together than the ground-truth, the model interprets the initial condition as deep interpenetration. As such conditions are inherently non-physical, any predicted behavior by the model is also necessarily non-physical and may be considered a *simulation artifact*. In particular, many simulators respond to interpenetration with normal impulses that scale exceedingly quickly with the depth (in the case of Stewart and Trinkle (1996) with slope $\sim \frac{1}{\Delta t}$; see the right side of Figure 5.1b and (5.2)), resulting in extremely large gradients $\|\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{prediction}}\|_2 >> 1$. Gradient descent in these regions is likely to be numerically challenging, and require exceedingly small step sizes to avoid missing the often-narrow region (in the 1D case, near $\hat{z}_g = z_g = 0$) where informative gradients could bring the training process to the true global optimum.

5.3. Suitability of Prediction Error for Stiff, Optimization-based Dynamics.

Given the difficulty of optimizing prediction error, we will seek in this chapter to develop an alternative model-fitting objective. In this section, we argue for a particular statistical analysis of model-fitting objectives, graph distance, which motivates low prediction error as a strong indicator of model quality. We then then use this analysis to qualify conditions by which alternatives to prediction error might achieve superior optimizability, while retaining a firm statistical interpretation.

Consider a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{u}_i, \boldsymbol{v}_i')\}$. For real-world robotic systems, the input $\boldsymbol{u}_i$ is driven by software decisions, and therefore often known exactly. However, the state information $\boldsymbol{x}_i, \boldsymbol{v}_i'$ is often reconstructed from noisy sensor measurements, and in general not exactly equal to the true underlying states. If the noise component of likely to be small (e.g., when $\boldsymbol{x}_i$ and $\boldsymbol{v}_i'$ have Gaussian white noise), then a necessary condition for datapoint i to match the model $\boldsymbol{f}_{\boldsymbol{\theta}}$ well is that there exists a pair $(\bar{\boldsymbol{x}}_i', \bar{\boldsymbol{v}}_i') \approx (\boldsymbol{x}_i, \boldsymbol{v}_i')$ such that $(\bar{\boldsymbol{x}}_i, \boldsymbol{u}_i, \bar{\boldsymbol{v}}_i')$ lies in the graph of $\text{Graph}(\boldsymbol{f}_{\boldsymbol{\theta}})$ of $\boldsymbol{f}_{\boldsymbol{\theta}}$:

$$\text{Graph}(\boldsymbol{f}_{\boldsymbol{\theta}}) = \left\{ \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \\ \boldsymbol{v}' \end{bmatrix} : \boldsymbol{v}' = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}) \right\}. \tag{5.3}$$

Graph distance squared (GDS) is therefore a natural metric to capture model accuracy:

$$\mathcal{L}_{\text{graph}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_i \text{Dist}\left([\boldsymbol{x}_i;\ \boldsymbol{v}_i'], \text{Graph}(\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot, \boldsymbol{u}_i, \cdot))\right)^2, \tag{5.4a}$$

$$= \frac{1}{|\mathcal{D}|} \sum_i \min_{\bar{\boldsymbol{x}}_i} \left\| \begin{bmatrix} \bar{\boldsymbol{x}}_i - \boldsymbol{x}_i \\ \boldsymbol{v}_i' - \boldsymbol{f}_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}_i, \boldsymbol{u}_i) \end{bmatrix} \right\|_2^2. \tag{5.4b}$$

It is often reasonable to believe there are similar noise distributions the current and next state, and thus weight the difference in $\boldsymbol{x}_i$ and $\boldsymbol{v}_i'$ equally in (5.4). By contrast, prediction error (2.49) only considers noise in the next state, replacing the minimization over $\bar{\boldsymbol{x}}_i$ with the constraint $\bar{\boldsymbol{x}}_i = \boldsymbol{x}_i$.

While GDS is a less ubiquitous performance metric than prediction error, it has a rich history in statistical analysis through *errors-in-variables* modeling (Cifarelli, 1988), dating as far back as a

method for linear regression proposed by Adcock (1878). One potential reason GDS-type objectives are not more widely used is the complexity introduced in the form of an optimization problem (5.4) embedded in the objective itself, turning the training process (2.48) into a complex, bilevel optimization problem. For non-stiff systems, it is not clear that this complexity is warranted; in fact, for $\boldsymbol{f_\theta}$ with a small Lipschitz constant, prediction error and GDS are equal up to a small constant:

Lemma 2. *If $\boldsymbol{f_\theta}$ is L_f-Lipschitz in $\boldsymbol{x}$, $\mathcal{L}_{\text{prediction}}(\boldsymbol{f_\theta}, \mathcal{D}) \geq \mathcal{L}_{\text{graph}}(\boldsymbol{f_\theta}, \mathcal{D}) \geq \frac{\mathcal{L}_{\text{prediction}}(\boldsymbol{f_\theta}, \mathcal{D})}{1+L_f^2}$.*

Proof. See Appendix C.1. □

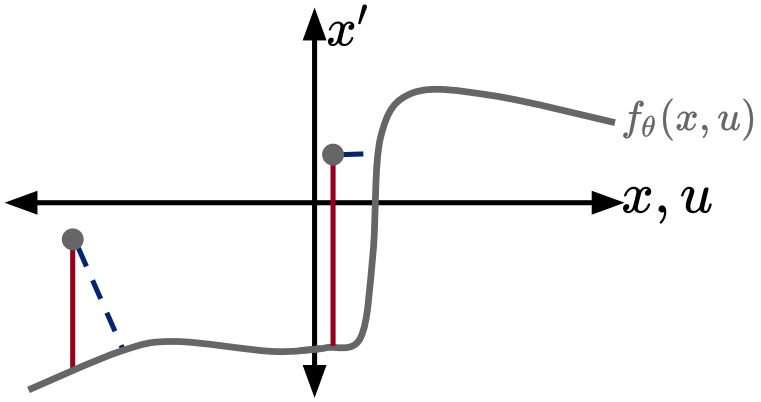


Figure 5.2: Illustration of the discrepancy between graph distance squared and prediction error, adapted from Bianchini et al. (2022). Geometrically, graph distance is measured between datapoints (gray dots) and the closest point on the model (blue dashed lines), whereas prediction error is the vertical distance to the model (red solid lines). For datapoints near a smooth/non-stiff region of the model (left), these two quantities are similar. By contrast, for datapoints near a stiff/nearly-vertical region of the model (right), a drastic disparity between these metrics occurs.

In particular, the upper bound on GDS in Lemma 2 does not depend on L_f. As the training process consists of minimizing $\mathcal{L}_{\text{prediction}}$, regardless of the particular value of L_f, low prediction error implies low graph distance, implying that the model well-explains the training data.

However, we find that stiff, optimization-based models, such as those for simulating rigid-body dynamics, obey a unique combination of properties that make prediction loss a less practically desirable tool for analysis. First, the model already embeds an optimization problem in prediction

loss. It is then unclear without analysis of the particular structure of the dynamics that the problem embedded in $\mathcal{L}_{\text{graph}}$ is more computationally difficult than forward prediction. Second, the stiffness of the model can induce a large discrepancy between prediction loss and graph distance, which scales with the square of the Lipschitz constant L_f of $\boldsymbol{f_\theta}$ (Lemma 2). This can result in large prediction error even when the model closely matches the data (see Figure 5.2 for a graphical explanation). As the Lipschitz constant of the 1-D system is $\frac{1}{\mathrm{d}t}$, the discrepancy is on the order of $\frac{1}{\mathrm{d}t^2} >> 1$ (see Figure 5.1b). In contrast to prediction error, the optimization landscape of GDS (shown in blue dashes) has large, smooth basin around the correct model. It therefore seems advantageous in this example, as well as rigid-body identification in general, to use an alternative approximation of GDS, which we hope could more closely inherit its optimization landscape.

5.4. An Alternative Objective via Constraint Violation Penalization

We now describe the inspiration and construction of a novel alternative objective to prediction error, and prove its relationship to GDS.

We turn to recent work in planning and control for robotic systems, where plans and controllers are reliably derived, despite similar gradient pathologies with respect to control inputs (Posa et al., 2013; Aydinoglu et al., 2023; Wensing et al., 2022). In such methods, one approach is to turn the optimization into a large, uni-level optimization problem, in which states; inputs; and contact forces are jointly optimized, subject to dynamics; input; and planning/control constraints. In such constructions, the contact constraints embedded in the dynamics may be temporarily replaced with a penalty or looser constraint. For instance, in Posa et al. (2013), the no-force-at-a-distance constraint $\boldsymbol{\Lambda}_{n,i}\boldsymbol{\phi}_i = 0$ is replaced with $\boldsymbol{\Lambda}_{n,i}\boldsymbol{\phi}_i \leq \varepsilon$; and in Aydinoglu et al. (2023), the constraint is replaced with a penalty during one alternation of ADMM on each solver iteration. As a result of this flexibility, these algorithms can reliably discover plans involving new contact modalities in their optimization process.

Our goal is to replicate this process in the system identification setting, by developing a model objective which effectively conducts an *inference* over contact modes at each gradient step. We note that the GDS indirectly achieves this goal due to its optimization over the initial condition

(5.4). However, by leveraging the optimization-based structure (2.37) of our dynamics of interest, we can construct an alternative objective $\mathcal{L}_{\text{violation}}$ which performs this inference explicitly by replacing the $h = 0$ constraint (see Assumption 3) with a penalty on its violation:

$$\mathcal{L}_{\text{violation}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_i \min_{\boldsymbol{\Lambda} \in \mathcal{K}} l_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{u}_i, \boldsymbol{v}_i', \boldsymbol{\Lambda}) \,, \tag{5.5a}$$

$$l_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}', \boldsymbol{\Lambda}) = \frac{1}{2} \left\| \boldsymbol{v}' - \boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\Lambda}) \right\|_2^2 + h_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}', \boldsymbol{\Lambda}) \,. \tag{5.5b}$$

In the particular case of rigid-body dynamics (2.39)–(2.40), the first term in this minimization can be thought of as analogous to prediction error, in which contact forces are partially inferred by minimization violation of the manipulator equations; and the second term is a violation of the complementarity and dissipation laws of contact.

We note that a unique insight into our construction is the use of the *datapoint's* next velocity $\boldsymbol{v}_i'$ in $h_{\boldsymbol{\theta}}$, rather than the model prediction $\boldsymbol{g}(\boldsymbol{x}_i, \boldsymbol{u}_i, \boldsymbol{\Lambda})$. As a result, unlike for forward prediction (2.38), the optimization problem is a tractable, convex, second-order cone program Boyd and Vandenberghe (2004), rather than a non-convex LCP. Consequently, this formulation has the potential to be less computationally burdensome than forward simulation and therefore prediction error.

In Figure 5.1b, we plot this loss on the 1D example (green dotted/dashed line), and we find that it appears to more tightly approximate GDS, and by virtue inherits a large basin of attraction to the true model parameters. Due to Assumption 3, we additionally know that this loss does not introduce unwanted bias in the sense that it's minimum value is 0, and that $\mathcal{L}_{\text{violation}} = 0$ if and only if $\mathcal{L}_{\text{graph}}$ and/or $\mathcal{L}_{\text{prediction}}$ are also 0. It is also clear on this particular plot and system that $\mathcal{L}_{\text{violation}} \geq \frac{1}{2}\mathcal{L}_{\text{graph}}$. Thus, in a similar manner to prediction error, low violation loss $\mathcal{L}_{\text{violation}}$ guarantees low GDS. It is not immediately clear however if outside of the window plotted, or for other, more complex systems, that $\mathcal{L}_{\text{violation}} \gtrsim \mathcal{L}_{\text{graph}}$. We conclude this section with a proof of such a bound in certain scenarios:

Theorem 1 (Violation loss bounds GDS)**.** *Assume the following:*

1. $\Delta t \leq \frac{1}{4}$[s].

2. $\boldsymbol{\Gamma}(\boldsymbol{q}) = \boldsymbol{I}$; *that is,* $\boldsymbol{v} = \dot{\boldsymbol{q}}$.

3. $\boldsymbol{J}(\boldsymbol{q})$ *and* $F_s(\boldsymbol{x}, \boldsymbol{u})$ *are constant in* $\boldsymbol{x} = [\boldsymbol{q}; \boldsymbol{v}]$.

4. $\boldsymbol{M}(\boldsymbol{q}) = \boldsymbol{I}$.

Let $\sigma_{\min}$ *be the minimum singular value of* $\boldsymbol{J}_n \boldsymbol{J}_n^T$ *and* $\sigma_{\max}$ *the maximum singular value of* $\boldsymbol{J}^T \boldsymbol{J}$. *Then for any dataset* $\mathcal{D}$ *and model parameters* $\boldsymbol{\theta}$ *for the rigid-body dynamics* $\boldsymbol{f}_{\boldsymbol{\theta}}$ *defined by* (2.38)–(2.39)*,*

$$\mathcal{L}_{\text{violation}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}) \geq k \mathcal{L}_{\text{graph}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \mathcal{D}) \,, \tag{5.6}$$

$$k = \frac{\sigma_{\min}}{8 \max(1, \sigma_{\max})} \,. \tag{5.7}$$

Proof. See Appendix C.3. □

We conclude this section with a discussion of the limitations of Theorem 1.

We first examine the four assumptions in the theorem statement. The first assumption in Theorem 1 is made for convenience in the proof process, and is simply that the timestep duration Δt is reasonably small; in typical robotics scenarios, Δt will be much smaller than the stated value. However, the later assumptions may be significantly limiting, and in general are violated by 3D articulated systems. Our current view is that these constraints aid in the proof process, and are not necessarily fundamental limitations in the loss. Future work may investigate relaxation of these assumptions in favor of norm bounds and smoothness properties of $\boldsymbol{\Gamma}$, F_s, and $\boldsymbol{M}$. For now, this theorem is used to practically guide the scaling of various terms in (2.40) inside the loss (5.5). Under this guidance, we will see that it performs well in both simulation and real-world scenarios. Additionally, the looseness factor k in the bound above is essentially the inverse of the conditioning of $\boldsymbol{J}$; in many cases in robotics, this term may be considerably less than 1.

It is worth noting that Theorem 1 only demonstrates that our violation loss is at least as large as

$\mathcal{L}_{\text{graph}}$. It does not guarantee that $\mathcal{L}_{\text{violation}}$ can't grow much larger than $\mathcal{L}_{\text{graph}}$. We do see that in at least some system, such as in the 1D plot shown in Figure 5.1b, the violation loss is often significantly less than prediction error, and has favorable practical qualities for optimization.

Finally, we note that the second-order cone program in (5.5) has identical dimension; constraints; and curvature to the convex approximation of Anitescu (2005) discussed in Section 2.6.1 (both have objectives with quadratic form $\frac{1}{2}\boldsymbol{\Lambda}^T\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T\boldsymbol{\Lambda}$), providing for nearly identical solving infrastructure and characteristics. Furthermore, as Anitescu's approximation has a similar representation as optimization-based dynamics (2.47) to Stewart and Trinkle (1996) (2.40), the violation loss can be easily adapted to this formulation. While we do not provide a detailed proof, we note that a directly equivalent result to Theorem 1 can be derived with nearly identical algebra.

5.5. Experimental Analysis

We now directly analyze the practical optimizability of our novel loss $\mathcal{L}_{\text{violation}}$ on a more complex 3D system. While our discussion has focused on characterizing the gradients of various losses with respect to their parameters, the ultimate purpose of this analysis to to hypothesize about the performance of (stochastic) gradient descent in effectively globally optimizing the parameters. To test this property directly, we again consider a simulated system, such that the true optimal parameters $\boldsymbol{\theta}^*$ are known exactly. We now discuss the details of the setup of our experimentation and the differing results.

5.5.1. Example System

We once again consider the die roll system similar to that Chapter 4, with the following changes:

- Contact simulation is perfectly rigid, and conducted by our own numerical implementation of Anitescu (2005), which is solved and differentiated with the method and software of Agrawal et al. (2019).
- The ground-truth physical parameters are exactly the hand-identified ones of the real system used in Pfrommer* et al. (2020); see Table 6.2.

- No noise is added to the data, such that the ground-truth parameters are the global optimum of both $\mathcal{L}_{\text{prediction}}$ and $\mathcal{L}_{\text{violation}}$.

We collect 512 training trajectories of 80 timesteps from a similar random initial conditions to Chapter 4, and similar to Section 4.3 split the dataset 50:50 by trajectories into hyperparameter optimization and evaluation sets.

5.5.2. Learning Process

We consider identification of the friction coefficient $\boldsymbol{\mu}$ and geometry of the die. The die is modeled as a convex polytope with 8 vertices $\boldsymbol{p}_i$, such that

$$\boldsymbol{\theta} = \begin{bmatrix} \hat{\boldsymbol{\mu}} & \hat{\boldsymbol{p}_1} & \dots & \hat{\boldsymbol{p}_8} \end{bmatrix} \in \mathbb{R}^{25} \,. \tag{5.8}$$

For each training instance, $\boldsymbol{\theta}$ is initialized in the range of 10-30% initial relative error, described precisely in Section 5.5.3.

Training via minimization of $\mathcal{L}_{\text{prediction}}$ and $\mathcal{L}_{\text{violation}}$ are compared over repeated trials. Trajectory sets are further split 50:25:25 into train, validation, and test sets, and trained with Adam (Kingma and Ba, 2015) with early stopping when the validation set loss no longer decreases. The learning rate and weight decay parameters used in these experiments come from brute-force optimizaiton for each method over 100 trials on the hyperparameter trajectory set via the Optuna library's (Akiba et al., 2019) implementation of the Hyperband optimizer (Li et al., 2018).

5.5.3. Performance Metric

At the conclusion of training, we check the relative parameter error between the identified $\boldsymbol{\theta}$ and the true friction coefficient geometry. As the true parameters $\boldsymbol{\theta}^*$ are the global minimizer of each loss, comparing the distance in parameter space to them provides a fair comparison between the optimizability of each loss.

The relative error for the friction coefficient is straightforwardly calculated as

$$e_{\boldsymbol{\mu}} = \frac{|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}|}{\boldsymbol{\mu}} \,. \tag{5.9}$$

For the geometric parameters, we consider the true volume occupied by the shape $\mathcal{S}$, as well as the learned shape volume $\hat{\mathcal{S}} = \mathrm{co}\left(\{\hat{\boldsymbol{p}}_i\}\right)$ implied by the learned vertices, and calculate a relative error volumetrically:

$$e_{\mathcal{S}} = \frac{\mathrm{Vol}\left((\mathcal{S} \setminus \hat{\mathcal{S}}) \cup (\hat{\mathcal{S}} \setminus \mathcal{S})\right)}{\mathrm{Vol}(\mathcal{S})} \,. \tag{5.10}$$

The total parameter error is then reported as $e_{rel} = \frac{1}{2}(e_{\boldsymbol{\mu}} + e_{\mathcal{S}})$.

5.5.4. Results

In Figure 5.3, we compare the values of e_{rel} at the initialization and conclusion of the model-fitting process. For each loss and model, the model-fitting process significantly decreased the relative error, showing the conceptual validity of both our loss and prediction error for recovering system parameters. However, in 13 out of 30 trials, prediction error failed to recover the true systems parameters to reasonable numerical precision ($e_{rel} = 1\%$). This supports our hypotheses that the gradient pathologies detailed in Section 5.2 lead to poor empirical system identification, even on noiseless data. By contrast, in every trial (30/30), our loss effectively globally optimized $\boldsymbol{\theta}$ to recover the die's true shape and coefficient of friction. This comparison supports our hypothesis that allowing the inference over modes in $\mathcal{L}_{\text{violation}}$ (5.5) allows for more effective fitting of stiff rigid-body models to data.

5.6. Discussion

The experimental results in Section 5.5 and theoretical results in Section 5.4 provide strong evidence that our novel loss $\mathcal{L}_{\text{violation}}$ is a well-reasoned objective for fitting physical model parameters to data. While the our results focused on rigid-body modeling, we note that many other physical models, such as some capturing solid-mechanical deformation, have a similar optimization-based structure (Han et al., 2023). Future work may investigate a formulation of $\mathcal{L}_{\text{violation}}$ to identify parameters of such systems, paired with similar theoretical results to Theorem 1.

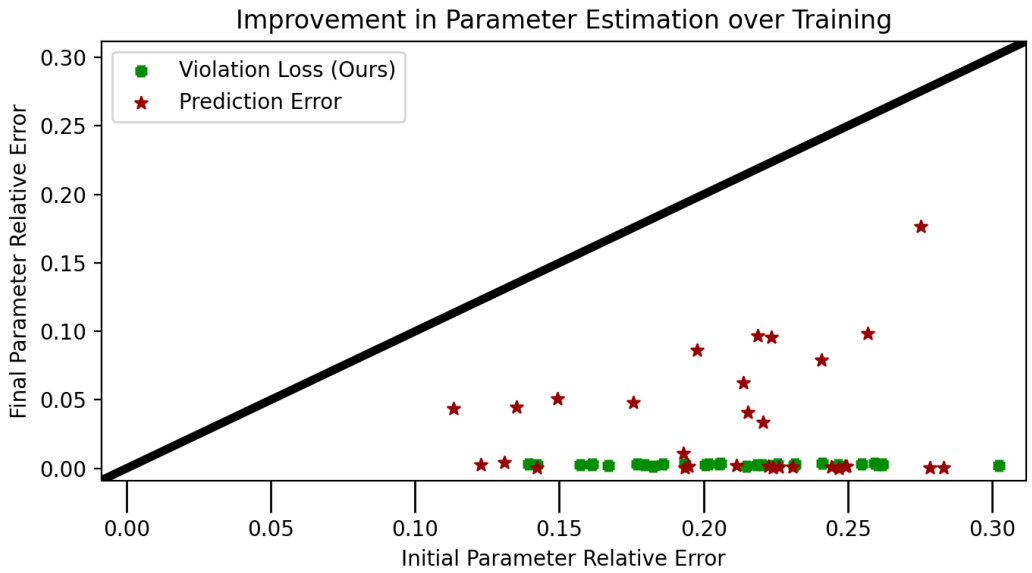


Figure 5.3: Initial and final relative parameter error is compared for fitting 3D die roll system parameters to date. We compare the effectiveness of optimizing via our novel loss $\mathcal{L}_{\text{violation}}$ (green) to a prediction error $\mathcal{L}_{\text{prediction}}$ baseline (red). In 30/30 trials, our loss recovered the true frictional and geometric properties of the cube to numerical precision (1%). By contrast, over half (16/30) of the prediction error-optimized trials recovered the true parameters, becoming stuck in local minima.

Additionally, our subsequent work Jin, Aydinoglu, Halm, and Posa (2022) examined applying a similar loss to $\mathcal{L}_{\text{violation}}$ for Linear Complementarity Systems, a local approximation of rigid-body models (Aydinoglu et al., 2023). These models can be used by robotics applications involving rigid-body systems that only require local accuracy, such as task-driven, model-based reinforcement learning (Jin and Posa, 2022).

Finally, the ultimate goal of our better-conditioned loss is to scale identification problems like that in Section 5.5 to the high-dimensional setting, e.g. identification in the space of all possible shapes. Our subsequent work will detail a method in pursuit of this goal by parameterizing geometries as deep neural networks.

CHAPTER 6

ContactNets: Learning Rigid Body Dynamics with Arbitrary Geometries via Implicit Neural Modeling

Parts of this chapter were previously published as parts of Samuel Pfrommer*, Mathew Halm*, and Michael Posa. ContactNets: Learning of Discontinuous Contact Dynamics with Smooth, Implicit Representations. In *The Conference on Robot Learning (CoRL)*, 2020. URL https://proceedings.mlr.press/v155/pfrommer21a.html. Samuel Pfrommer conducted the real-world data collection described in Section 6.4, and the remaining reproduced portions represent original contributions.

6.1. Introduction

As we discussed in Chapter 1, our results in Chapter 5 as well as prior study (Heiden et al., 2020; Fazeli et al., 2017b; Kolev and Todorov, 2015) show that a small number of parameters of rigid-body models can be effectively fit to data. However, in robotics we would like to build entire models from scratch, automatically, from limited data. In Chapter 4, we saw that while purely emirical DNN-based models in theory have the flexibility to accomplish this, in practice they are hampered by poor numerics and inductive biases. In this chapter, we combine our previously-discussed research with a novel, implicit representation to take a significant step towards building rigid-body models from scratch. Specifically, we seek to learn arbitrary object geometries without significant priors from small datasets. This setting is difficult, as the parameter space of geometries is infinite-dimensional in general. A successful method of doing so can enable robotic manipulators to quickly understand and utilize rigid and articulated objects in their environments.

In Section 6.2, we discuss how DNN's can be used in a "white-box" manner to paramterize arbitrary object geometries. Taking inspiration from the convex-decomposition representation embedded in common robotics simulators, we parameterize object shapes implicitly as a union of convex shapest. Each convex part is capture as a DNN modeling its support function, which is continuous and therefore well-utilizes DNNs' capabilities to perform smooth extrapolation. We derive a new, novel DNN architecture to this end, H-FICNN's, which we rigorously prove to universally approximate any

shape. In Section 6.3, we detail *ContactNets*, our method of embedding and training such DNN's inside rigid-body models, such that they can be made compatible with state-of-the-art robotic simulation, planning, and control environments. In Section 6.4, we demonstrate *ContactNets*' ability to recover high-quality models from real-world motion capture data of a die-roll system similar to the one explored in previous chapters. Finally in Section 6.5, we discuss the implications of our results and future directions for research.

6.2. Representing Arbitrary Shapes with Deep Neural Networks

In this section, we establish a theoretically-proven method of approximating arbitrary shapes as neural networks. We consider a bounded shape $\mathcal{S} \subseteq \mathbb{R}^3$. The problem which we would like to solve is to *implicitly* parameterize the shape with a DNN with a finite-dimensional basis. That is, we would like to define a family of neural networks $s_{\boldsymbol{\theta}} : \mathbb{R}^n \to \mathbb{R}$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ such that each selection of $\boldsymbol{\theta}$ defines an $s_{\boldsymbol{\theta}}(\cdot)$ that corresponds exactly to a single learned shape $\mathcal{S}_{\boldsymbol{\theta}} \subseteq \mathbb{R}^3$. We desire this representation to have the following properties:

1. There must be an efficient way to compute mesh approximations of the learned shape, a format compatible with many robotics applications including popular simulation and visualization environments.

2. The representation should be continuous with a reasonably-small Lipschitz constant for arbitrary shapes, such that the smooth interpolative power of DNN's is well-utilized.

3. The representation must have sufficient flexibility to approximate *any* shape. That is, for some neural network $s_{\boldsymbol{\theta}}$, there exist parameters $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ such that the learned shape $\mathcal{S}_{\boldsymbol{\theta}}$ and true shape $\mathcal{S}$ have arbitrarily-small discrepancy.

We now show how a concept from convex analysis, *support functions*, can satisfy each of these properties.

6.2.1. Convex Polytopes as Support Functions

We first consider the case where $\mathcal{S} = \text{co}(\{\boldsymbol{p}_i : i \in 0, \ldots, M-1\})$ is a convex polytope with a finite set of vertices $\boldsymbol{p}_i \in \mathbb{R}^3$. Such shapes are implicitly parameterized by their support function $s : S_2 \to \mathbb{R}$ (Bazaraa et al., 2006):

$$s(\hat{\boldsymbol{r}}) = \max_i \boldsymbol{p}_i \cdot \hat{\boldsymbol{r}} \,. \tag{6.1}$$

$s(\hat{\boldsymbol{r}})$ implicitly defines a hyperplane-intersection representation of $\mathcal{S}$:

$$\mathcal{S} = \bigcap_{\|\hat{\boldsymbol{r}}\|_2=1} \{\boldsymbol{p} : \hat{\boldsymbol{r}} \cdot \boldsymbol{p} \leq s(\hat{\boldsymbol{r}})\} \,. \tag{6.2}$$

The gradient[5] of s is equal to a *support point* of the shape (Bazaraa et al., 2006). This point is the element $\boldsymbol{p} \in \mathcal{S}$ that is farthest in the $\hat{\boldsymbol{r}}$ direction—i.e., $\boldsymbol{p} \cdot \hat{\boldsymbol{r}} = s(\hat{\boldsymbol{r}})$. These points can be used to construct a convex hull representation of the polytope:

$$\nabla_{\hat{\boldsymbol{r}}} s(\hat{\boldsymbol{r}}) = \arg\max_{\boldsymbol{p}_i} \boldsymbol{p}_i \cdot \hat{\boldsymbol{r}} \,, \tag{6.3}$$

$$\mathcal{S} = \text{co}\left(\bigcup_{\|\hat{\boldsymbol{r}}\|_2=1} \nabla_{\hat{\boldsymbol{r}}} s(\hat{\boldsymbol{r}}) \right) \,. \tag{6.4}$$

The relationship between $\mathcal{S}$, $\hat{\boldsymbol{r}}$, $s(\hat{\boldsymbol{r}})$ and $\nabla_{\hat{\boldsymbol{r}}} s(\hat{\boldsymbol{r}})$ is illustrated in Figure 6.1.

Equations (6.2) and (6.4) together constitute a computational framework to approximately extract $\mathcal{S}$ via sampling $s(\hat{\boldsymbol{r}})$ and its gradient. Such operations are computationally tractable and highly parallelizable for most DNN architectures. By randomly sampling unit directions $\hat{\boldsymbol{r}}_1, \ldots, \hat{\boldsymbol{r}}_K$, replacing the infinite intersection and union in (6.2) and (6.4) with finite ones results in outer- and inner-approximations $\mathcal{S}_{\text{outer}} \supseteq \mathcal{S} \supseteq \mathcal{S}_{\text{inner}}$. The resulting shapes are a finite intersection of hyperplanes and the convex hull of a finite vertex collection. These are standard "H-" and "V-" representations from which surface and volume meshes can be computed with standard libraries, e.g. `qhull` (Barber et al., 1996). The error in this approximation of $\mathcal{S}$ can then be quantified by inspection of

[5] s is in general not differentiable; for geometries with sharp corners, many points may be equally far in the $\hat{\boldsymbol{r}}$ direction. In this case, the same relationship can be made with the Fréchet subgradient of $s(\hat{\boldsymbol{r}})$ (Bazaraa et al., 2006). For more details on subgradient calculus, see Appendix A.2.1.

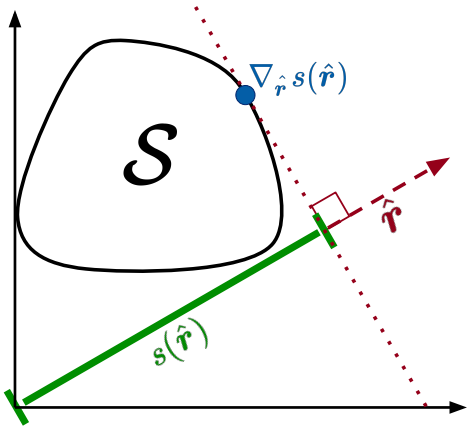


Figure 6.1: Illustration of the support function of a convex shape $\mathcal{S}$. Given an outward unit normal $\hat{\boldsymbol{r}}$, $s(\hat{\boldsymbol{r}})$ is how far the shape extends in this direction. $\boldsymbol{p} = \nabla_{\hat{\boldsymbol{r}}} s(\hat{\boldsymbol{r}})$ is equal to the point on the shape's boundary that lies as far as possible in this direction. We then have that $\boldsymbol{p} \cdot \hat{\boldsymbol{r}} = s(\hat{\boldsymbol{r}})$.

$\mathcal{S}_{\text{outer}} \setminus \mathcal{S}_{\text{inner}}$. It is also clear from (6.3) that $s(\hat{\boldsymbol{r}})$ is continuous with Lipschitz constant $\max_i \|\boldsymbol{p}_i\|_2$.

6.2.2. Support Function Approximation with Input-Convex Neural Networks

We have seen in the previous section that for convex polytopes, support functions satisfy the first two desired properties for our shape representation. We now demonstrate how support functions can be represented as neural networks. Support functions have an equivalence property with positively homogenous convex functions (Schneider, 1993). That is:

- Any positively homogeneous and convex function is the support function of a unique compact-convex shape.
- Every compact-convex shape's support function is positively homogeneous and convex.

Therefore, if convexity and positive homogeneity are guaranteed for a given neural network architecture, then any learned parameters will correspond to a unique compact-convex shape. We propose such an architecture, which extends the Fully-connected Input-Convex Neural Network (FICNN) architecture presented in Amos et al. (2017). FICNN's are convex in their inputs (Amos et al., 2017, Proposition 1). They achieve this with a similar structure to the simple feedforward network discussed in (2.51), with the following modifications:

- The hidden-layer weight matrices $\boldsymbol{W}_{\boldsymbol{z},k}$ must have non-negative entries. This can be done for instance by parameterizing them with the element-wise logarithms of their values, i.e. $\boldsymbol{W}_{\boldsymbol{z},k} = \exp(\bar{\boldsymbol{W}}_{\boldsymbol{z},k})$.

- The activation function $\boldsymbol{\sigma}(\cdot)$ is element-wise convex and monotonically increasing. For instance the popular (Leaky) rectified linear unit

$$\text{LeakyReLU}(\boldsymbol{y})_i \begin{cases} \boldsymbol{y}_i & \boldsymbol{y}_i \geq 0\,, \\ \alpha \boldsymbol{y} & \boldsymbol{y}_i < 0\,, \end{cases} \tag{6.5}$$

  with (constant) leaky fraction $\alpha \in [0, 1)$ satisfies this property. The simplified case $\alpha = 0$ is called $\text{ReLU}(\boldsymbol{y}) = \text{pos}(\boldsymbol{y})$.

In this work, we propose two additional constraints, which turn the FICNN layers into a nested sequence of positively homogeneous functions:

- The activation $\boldsymbol{\sigma}$ is also positively homogeneous; (Leaky)ReLU activations also have this property.

- Each bias $\boldsymbol{b}_k$ is set to zero.

We call networks meeting this full set of constraints a Homogenous FICNN (H-FICNN). As the composition of positively homogeneous functions is also positively homogeneous, each H-FICNN meets all structural requirements to be the support function of a unique compact-convex shape. We now show by proof that the converse is true: any convex polytope's support function can be exactly represented as an H-FICNN with ReLU activations.

Theorem 2. *Consider a bounded convex polytope* $\mathcal{S} = \text{co}\left(\{\boldsymbol{p}_i : 0 \in 1, \ldots, M-1\}\right)$ *with support function* $s(\hat{\boldsymbol{r}})$*. Then, there exists an H-FICNN* $s_{\boldsymbol{\theta}}(\hat{\boldsymbol{r}})$ *with width and depth* M *and ReLU activations such that* $s_{\boldsymbol{\theta}}(\hat{\boldsymbol{r}}) = s(\hat{\boldsymbol{r}})$ *everywhere and thus* $\mathcal{S}_{\boldsymbol{\theta}} = \mathcal{S}$*.*

Proof. See Appendix D.1. □

Along with the discussion in Section 6.2.1, this proof guarantees that support functions represented as H-FICNN's meet all three of the desired properties for shape representation in the domain of convex polytopes.

We not that another possibility would simply be to represent the polytope directly as the vertices $\{\boldsymbol{p}_i\}$. We find H-FICNN parameterizations advantageous in that they can achieve a more dense representation of complex geometries. For a parameterization $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ that can represent up to M vertices, we define its density as $\frac{M}{n_\theta}$. For instance, representing the polytope directly as M vertices requires $3M$ parameters, and thus has density $\frac{1}{3}$. In comparison, the approximately M^3 parameters of the network in Theorem 2 seems to imply that H-FICNNs are a less dense representation. However, universal approximation theorems like Theorem 2 don't necessarily represent the true limits on complexity of the underlying architecture. In fact, it is known that for fully connected DNN's with ReLU activations of input dimension 3; width $3w$; and depth $d < 3w$, the parameter density is at least $\frac{w^{3(d-1)}}{9}$ (Montúfar et al., 2014). In practice, we therefore opt for network widths and depths in the ranges $[64, 256]$ and $[2, 4]$, respectively. When embedded in rigid-body models, we find that networks of such size are evaluated quickly enough to make negligible contributions to the total computational burden of simulation. At the same time, the size of these networks allow them to approximate essentially any polytope, and thus it is not necessary to know or hard-code the number of vertices M beforehand.

6.2.3. Shape Approximation as Convex Polytope Decomposition

We have shown that H-FICNN's meet all of our desired properties for exactly representing convex polytopes. We now extend this idea to generic bounded shapes.

It is well known that any bounded shape $\mathcal{S}$ can be approximately decomposed into a finite collection of convex shapes $\mathcal{S}_1, \ldots, \mathcal{S}_N$. Such decompositions can be computed to arbitrary precision:

Theorem 3. *For any $D, \varepsilon > 0$ and compact shape $\mathcal{S} \subseteq [-D, D]^3$, there exists a finite collection of*

convex polytopes $\bar{\mathcal{S}} = \cup_{i \in 1,\dots,N} \mathcal{S}_i$ *such that*

$$\max \left(\max_{\boldsymbol{p} \in \mathcal{S}} \text{Dist}(\boldsymbol{p}, \bar{\mathcal{S}}), \max_{\bar{\boldsymbol{p}} \in \bar{\mathcal{S}}} \text{Dist}(\bar{\boldsymbol{p}}, \mathcal{S}) \right) < \varepsilon \,. \tag{6.6}$$

Proof. See Appendix D.2. □

It immediately follows that any compact shape can be arbitrarily approximated by a finite collection of H-FICNN's.

6.3. Embedding Learned Support Functions into Rigid Body Simulation in *ContactNets*

The previous section described a novel, theoretically-proven method to represent arbitrary shapes as DNN's. In this section we motivate and detail *ContactNets*, our proposed method by which these representations can be embedded into physics simulation; an overview of the process involved is given in Figure 6.2.

6.3.1. Simulating Contact with Support Functions

When popular simulation environments like Bullet (Coumans, 2015) import non-convex shapes for collision geometries, they opt to approximate them with a collection of convex shapes, in exactly the sense described in Theorem 3 Mamou and Ghorbel (2009). A primary motivation for this computation step is, in fact, to gain access to the support functions $s(\hat{\boldsymbol{r}})$ of each convex. The robotics and video game communities has for decades relied on collision detection and signed distance $\boldsymbol{\phi}(\boldsymbol{q})$ calculation algorithms, seminally including the method of Gilbert, Johnson, and Keerthi (1988) (GJK) and the Expanding Polytope Algorithm (EPA) (Van den Bergen, 2001), which internally represent shapes by their support functions. These methods take as input the relative rotation and translation $(\boldsymbol{R}^1_2, \boldsymbol{t}^1_{1,2})$ between two bodies defined by shapes $\mathcal{S}_1, \mathcal{S}_2$; and interfaces to compute the body-frame support function $s_k(\hat{\boldsymbol{r}})$ and support points $\boldsymbol{p}_k = \nabla_{\hat{\boldsymbol{r}}} s_k(\hat{\boldsymbol{r}})$. As central subroutines to countless robotics; gaming; and graphics algorithms, a large quantity of work has been conducted on maximizing accuracy and efficiency of the approximate convex decomposition (Mamou and Ghorbel, 2009; Mamou, 2016); collision detection (Liu et al., 2008; Wei et al., 2022); and signed-distance calculation (Fares and Hamam, 2009) process inside simulation. As a result, it is known that the

contact dynamics even of pathologically concave objects like bowls can be reliably and quickly simulated as unions of convex shapes.

We briefly describe how these algorithms lead to a construction of the contact geometry terms $\boldsymbol{\phi}_i(\boldsymbol{q}), \boldsymbol{J}_i$. Collision detection algorithms output a pair of world-frame (w) *witness points* $\bar{\boldsymbol{p}}_1^w(\boldsymbol{\theta}), \boldsymbol{p}_2^w(\boldsymbol{\theta})$ representing the closest pair of points on the bodies (if they are not touching) or the deepest pair of interpenetrating points (if the bodies occupy the same space); as well as the outward normal $\hat{\boldsymbol{n}}^w(\boldsymbol{\theta})$ of $\mathcal{S}_1$ at $\bar{\boldsymbol{p}}_1^w(\boldsymbol{\theta})$. We note that these quantities are dependent on the shape parameters $\boldsymbol{\theta}$. The contact dynamics terms $\boldsymbol{\phi}_{i,\boldsymbol{\theta}}(\boldsymbol{q}), \boldsymbol{J}_{i,\boldsymbol{\theta}}(\boldsymbol{q})$ are algorithmically constructed using the following steps:

1. Calculate $\boldsymbol{\phi}_{i,\boldsymbol{\theta}}(\boldsymbol{q}) = \hat{\boldsymbol{n}}^w(\boldsymbol{\theta}) \cdot (\bar{\boldsymbol{p}}_2^w(\boldsymbol{\theta}) - \bar{\boldsymbol{p}}_1^w(\boldsymbol{\theta}))$.

2. Construct the rotation matrix between the contact frame c and world frame $\boldsymbol{R}_w^c(\boldsymbol{\theta})$ such that the first row ($\hat{\boldsymbol{x}}_c^w$-axis) is $\hat{\boldsymbol{n}}^w(\boldsymbol{\theta})$.

3. Using forward kinematics (which does not directly depend on the collision geometry), calculate $\boldsymbol{J}_k^w(\bar{\boldsymbol{p}}_k^w(\boldsymbol{\theta})) \in \mathbb{R}^{3 \times n_v}$, the world frame velocity Jacobian of $\frac{\mathrm{d}}{\mathrm{d}t}\bar{\boldsymbol{p}}_k^w(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{v}$.

4. Calculate the contact Jacobian as $\boldsymbol{J}_{i,\boldsymbol{\theta}}(\boldsymbol{q}) = \boldsymbol{R}_w^c(\boldsymbol{\theta})(\boldsymbol{J}_2^w(\bar{\boldsymbol{p}}_2^w(\boldsymbol{\theta})) - \boldsymbol{J}_1^w(\bar{\boldsymbol{p}}_1^w(\boldsymbol{\theta})))$.

6.3.2. Training and utilizing learned geometries with *ContactNets*

We now detail *ContactNets*, our method of fitting arbitrary geometries in rigid-body models to to data.

Input Data

As in typical empirical system identification (Section 2.7), we consider identification of rigid-body dynamics parameters from a finite collection of state trajectories $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}')_i\}$. For real-world systems, such information may be recorded for instance via a combination of proprioception and motion capture.

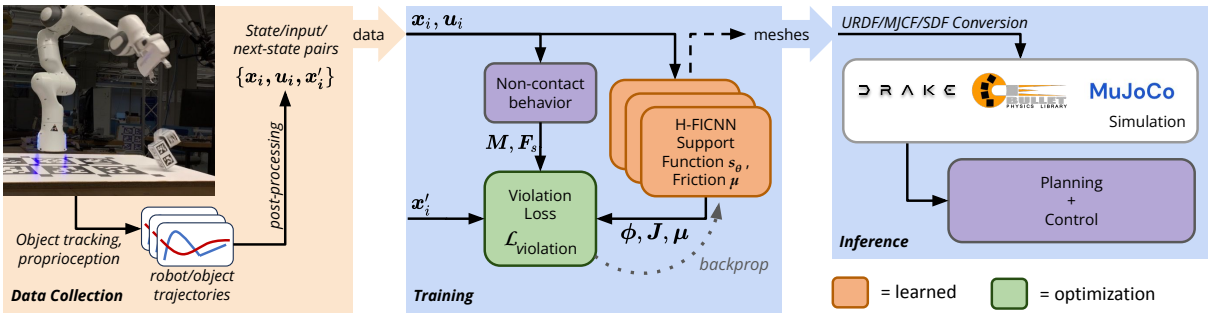


Figure 6.2: Overview of the proposed method for constructing physics-based models from state trajectories. **Left**: Motion capture data is collected from real-world objects as they move through the environment. **Center**: geometries of objects are parameterized as H-FICNN's representing their convex decomposition. These networks are fit to data via the mechanics-inspired violation loss $\mathcal{L}_{\text{violation}}$ discussed in Chapter 5. **Right**: The geometries can then be extracted to a simulator configuration file, such as a URDF. This file can then be loaded by various robotic simulation, planning, and control environments.

Model Representation

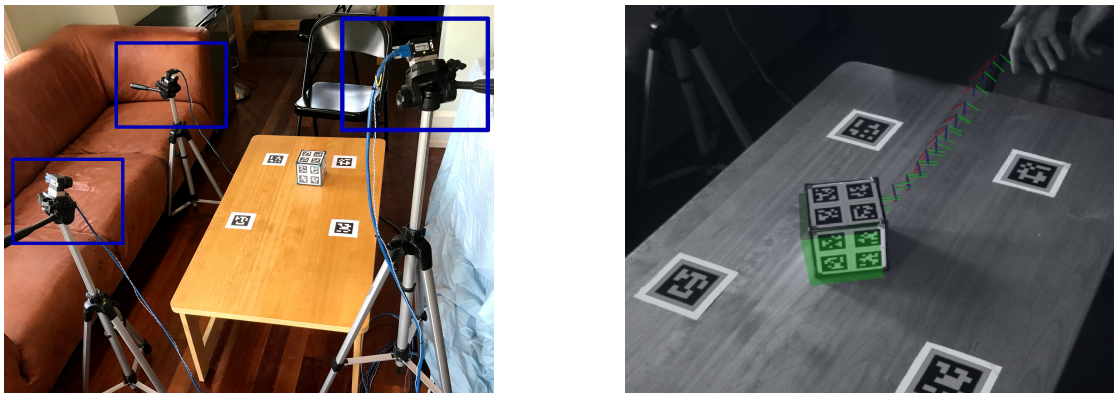
Topological and discrete information such as the number of bodies in the environment and robot joint arrangement are assumed to be known *a priori*. Typical scalar-, vector-, and tensor-valued quantities like mass; link-lengths; coefficients of friction; and inertia may be parameterized with standard methods, e.g. Rucker and Wensing (2022). Unknown geometries are represented as unions of convex shapes, which are parameterized as H-FICNN's; the number of convexes in each decomposition is fixed. While learning inertial quantities is not explored in this text, our recent extensions of *ContactNets* have successfully learned inertial and contact terms simultaneously (Bianchini, Halm, and Posa, 2023).

Training and Inference

Models are fit to data using via stochastic descent on the violation loss $\mathcal{L}_{\text{violation}}$ derived in Chapter 5. In each of our experiments, we used the Adam optimizer (Kingma and Ba, 2015). Once a set of parameters are fit, because even our novel deep-learned geometries may be converted to standard meshes, the learned model may then be exported as a URDF, MJCF, or SDF file. These representations may then be imported by the most ubiquitous robotics simulators, including Drake (Tedrake, 2014), Bullet (Coumans, 2015), and MuJoCo (Todorov, 2014). Futuremore, popular plan-

ners (Posa et al., 2016a) and controllers (Aydinoglu et al., 2023) for rigid-body system with contact can accept URDF models via the Drake library, allowing the learned model to be used directly by these algorithms as well.

6.4. Experiments



(a) (b)

Figure 6.3: Our real-world system designed to replicate the complex, three dimensional contact explored in Chapters 4 and 5. (a): A cube was tossed against the planar ground, producing impacts and both sticking and sliding frictional contact. Cube position and orientation are tracked by three cameras, outlined in blue. (b): A trace of c.o.m. position and block orientation is superimposed on a typical tossing behavior learned by our method, *ContactNets*. In both real and predicted (green) behavior, the block impacts the ground on a corner and tumbles onto its side before coming to rest.

The ultimate goal of our modeling methods is to be able to construct accurate models of real-world systems from as little data as possible. We thus choose in these experiments to evaluate the abilities of our method on a representative real-world canonical system. Specifically, we test *ContactNets*' ability to model a real-world object's dynamics by reconstructing its geometry from motion capture data. Beyond the benefits of being compatible with state-of-the-art robotics planners and controllers, we hypothesize that both the physics-based modeling structure as well as the use of our novel violation loss $\mathcal{L}_{\text{violation}}$ aid in finding an accurate dynamical model from less data. To test this, we conduct an ablation study on these components. Over varying dataset sizes, we compare the long-term predictive ability of learned models on a real-world analogue of the die rolling system explored in previous chapters.

Table 6.1: Examined Model-fitting Methods

Method Name	**Model Parameterization**	**Training Loss**
ContactNets (ours)	Rigid-Bodies w/ DNN Shapes (ours)	$\mathcal{L}_{\text{violation}}$ (5.5) (ours)
Deep Differentiable Physics (ours)	Rigid-Bodies w/ DNN Shapes (ours)	$\mathcal{L}_{\text{prediction}}$ (2.49)
End-to-End	Fully-learned DNN $\boldsymbol{f_\theta}(\boldsymbol{x}, \boldsymbol{u})$	$\mathcal{L}_{\text{prediction}}$ (2.49)

6.4.1. Compared Methods

We compare the predictive performance of the following three model-fitting methods, summarized in Table 6.1:

- ***ContactNets*** (ours): This method is the full proposed method, which represents shapes as unions of convex H-FICNN's; and is trained by minimization of our novel loss $\mathcal{L}_{\text{violation}}$.

- ***Deep Differentiable Physics*** (ours): This method is the same novel shape parameterization as *ContactNets*, but follows standard practice of training to minimize prediction error $\mathcal{L}_{\text{prediction}}$. Given the poor landscape of $\mathcal{L}_{\text{prediction}}$ described in Chapters 4 and 5, we expect this method to struggle with local minima, and consistently converge to poor model parameters regardless of the dataset size.

- ***End-to-end***: This method follows the standard practice to learn the mapping $\boldsymbol{x}' = \boldsymbol{f_\theta}(\boldsymbol{x}, \boldsymbol{u})$ entirely as a single DNN, in this case an MLP. We expect this method to suffer from the same stiffness-induced challenges explored in Chapter 4.

6.4.2. Example System

In theme with Chapters 4 and 5, we consider as real analogue to the "die roll" system, shown in Figure 6.3a. A 10 [cm] acrylic cube is tossed onto a wooden surface, which produces nearly instantaneous (i.e., sub-timestep) impact behaviors, including tumbling, sliding, and multiple impacts in a single timestep. In order to protect the cube from damage over many impacts, each corner is covered with a thin film of soft material. Physical paramters of this system are listed in Table 6.2. While real world systems often have frictional behaviors that diverge significantly from Coulomb friction, we provide a least-squares fit to the kinetic coefficient of friction to provide coarse intuition

Table 6.2: Real-world Die Roll System Parameters

Constant	Symbol	Value	Units
mass	m	0.37	kg
inertia	I	8.1×10^{-4}	$\mathrm{kg\,m^2}$
side length	l	0.1048	m
gravity	g	9.81	$\frac{\mathrm{m}}{\mathrm{s}^2}$
friction coefficient	μ	0.208	(none)
time-step	Δt	6.8×10^{-3}	s

about the contact behavior.

Each side of the cube features four unique AprilTags tracked by three 148 [Hz] PointGrey cameras using TagSLAM (Pfrommer and Daniilidis, 2019). Object configurations are recorded as $[\boldsymbol{t};\ \boldsymbol{R}]$ consisting of the world-frame center of mass position $\boldsymbol{t}$ and rotation matrix $\boldsymbol{R}$, and velocities are constructed by finite difference. 550 high-quality tosses are recorded.

6.4.3. Learning Process

For the rigid-body models, the identified parameters are the coefficient of friction $\boldsymbol{\mu}$ and geometry of the die. The die is modeled as a single convex shape, with parameterization as an H-FICNN with width 256 and depth 2. For each training instance, $\boldsymbol{\theta}$ is initialized in the range of 10-30% initial relative error, described previously in Section 5.5.3. 256 tosses are reserved for hyperparameter optimization. The learning rate; weight decay; and minibatch size parameters are brute-force optimized for each method over 100 trials via Hyperband (Li et al., 2018), in the same manner as in Chapter 5. The MLP's in the *End-to-end* models additionally have their width and depth optimized by the same process.

The remaining trajectory sets are further split 50:25:25 into train, validation, and test sets. All models are trained with Adam (Kingma and Ba, 2015) and early stopping when the validation set loss no longer decreases.

Inside each loss $\mathcal{L}_{\text{prediction}}$, $\mathcal{L}_{\text{violation}}$, there is a prediction component scaling with $\|\boldsymbol{v}\|_2^2$. We take note that there is an implicit mixing of linear and angular units here that may lead to poor numerical

scaling. We find that superior performance can be achieved when positions are normalized by a *characteristic length* similar to the side length of the cube. In these experiments, we specifically use $l_{\text{char}} = \sqrt{\frac{I}{m}} = 4.68\,[\text{cm}]$[6]. This is equivalent to scaling the velocity prediction error as $\|\boldsymbol{v}\|^2_{\boldsymbol{M}(\boldsymbol{q})}$.

6.4.4. Performance Metric

While the previous study in Chapter 5 examined ground-truth parameter error, this study faces two issues in applying the same metric. The first is that *End-to-end* models are non physically structured; and the second is that some parameters like the Coulomb friction component do not have a real value. We therefore examine empirical accuracy of the methods as their metric of quality. Given the length scaling described in Section 6.4.3, we propose to evaluate the long-term prediction of the position of the cube in these units. For each trajectory in the training and test sets, we initialize each model to the ground-truth initial condition; simulate forward in time to record $[\hat{\boldsymbol{t}}_n, \hat{\boldsymbol{R}}_n]$ at each time n; and calculate the time-averaged discrepancy with the ground truth $[\boldsymbol{t}_n, \boldsymbol{R}_n]$:

$$e_{\text{pred}} = \frac{1}{N}\sum_{n=1}^{N} |\text{angle}(\hat{\boldsymbol{R}}_n, \boldsymbol{R}_n)| + \frac{1}{l_{\text{char}}} \left\|\hat{\boldsymbol{t}}_n - \boldsymbol{t}_n\right\|_2 \,[\text{rad}]\ . \tag{6.7}$$

Above, $\text{angle}(\hat{\boldsymbol{R}}_n, \boldsymbol{R}_n)$ is the angle of the angle-axis rotation from $\hat{\boldsymbol{R}}_n$ to $\boldsymbol{R}_n$.

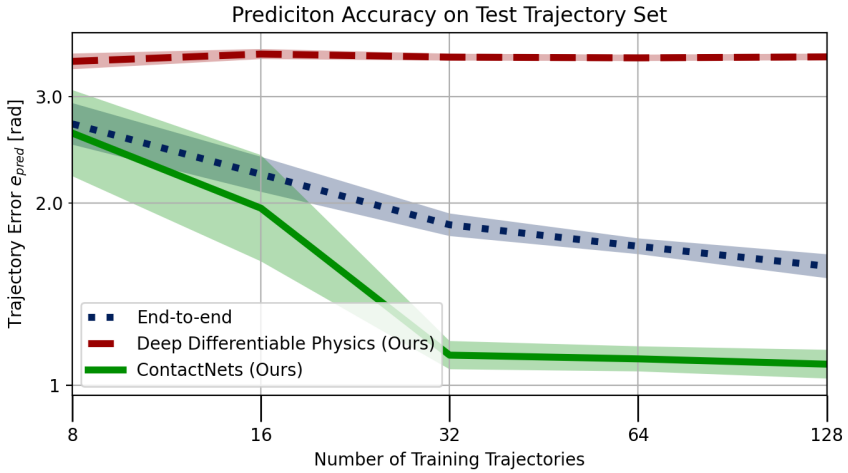
The value e_{pred} takes over the test trajectory set is our primary metric of predictive performance for each model. We additionally quantify e_{pred} over the training set as well to aid in understanding why some methods have better test set performance than others.

6.4.5. Results

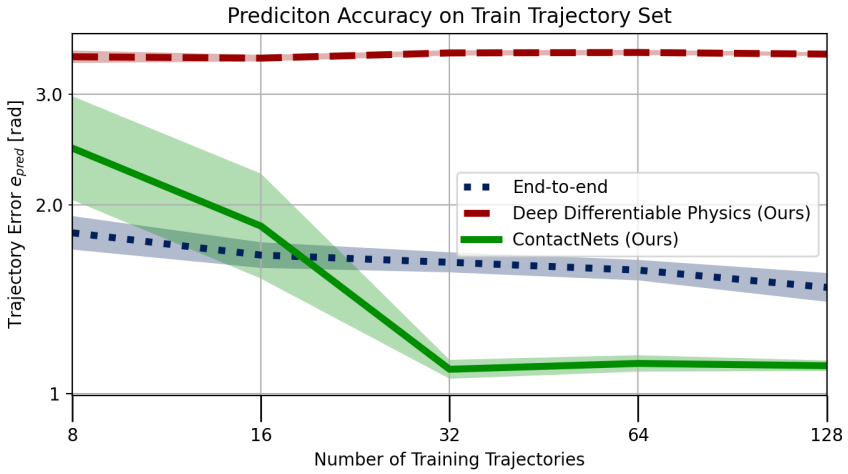
In Figure 6.4, we compare the long-term prediction error e_{pred} between *ContactNets* and our baseline methods across several dataset sizes. Similar to Chapter 4, we overlay log-normal 95% confidence intervals, estimated from 30 trials per method and dataset size.

We find that with only 32 training tosses tosses, *ContactNets* reliably and significantly outperforms both baselines. By comparison, *Deep Differentiable Physics* models have the poorest training and test set errors, and do not significantly improve with added data. In Figure 6.5, for both types of

[6]The intuition in this scaling is that inertia scales as $\sim ml^2$



(a)



(b)

Figure 6.4: We compare the long-term test-set (a) training-set (b) prediction error e_{pred} between *ContactNets* and two baseline methods, with overlayed 95% confidence intervals. (Green line): with only 32 training tosses tosses, *ContactNets* reliably and significantly outperforms both baselines. (Red dashes): By comparison, training the same model with prediction error in *Deep Differentiable Physics* results in the training process becoming stuck in local minima, and poor performance regardless of dataset size. (Blue dots): *End-to-end* perform similarly or worse than *ContactNets* on test data, especially in the middle-to-high data regime. These models have by far the largest discrepancy in performance between the training set and test set, meaning that their generalization error is largest.

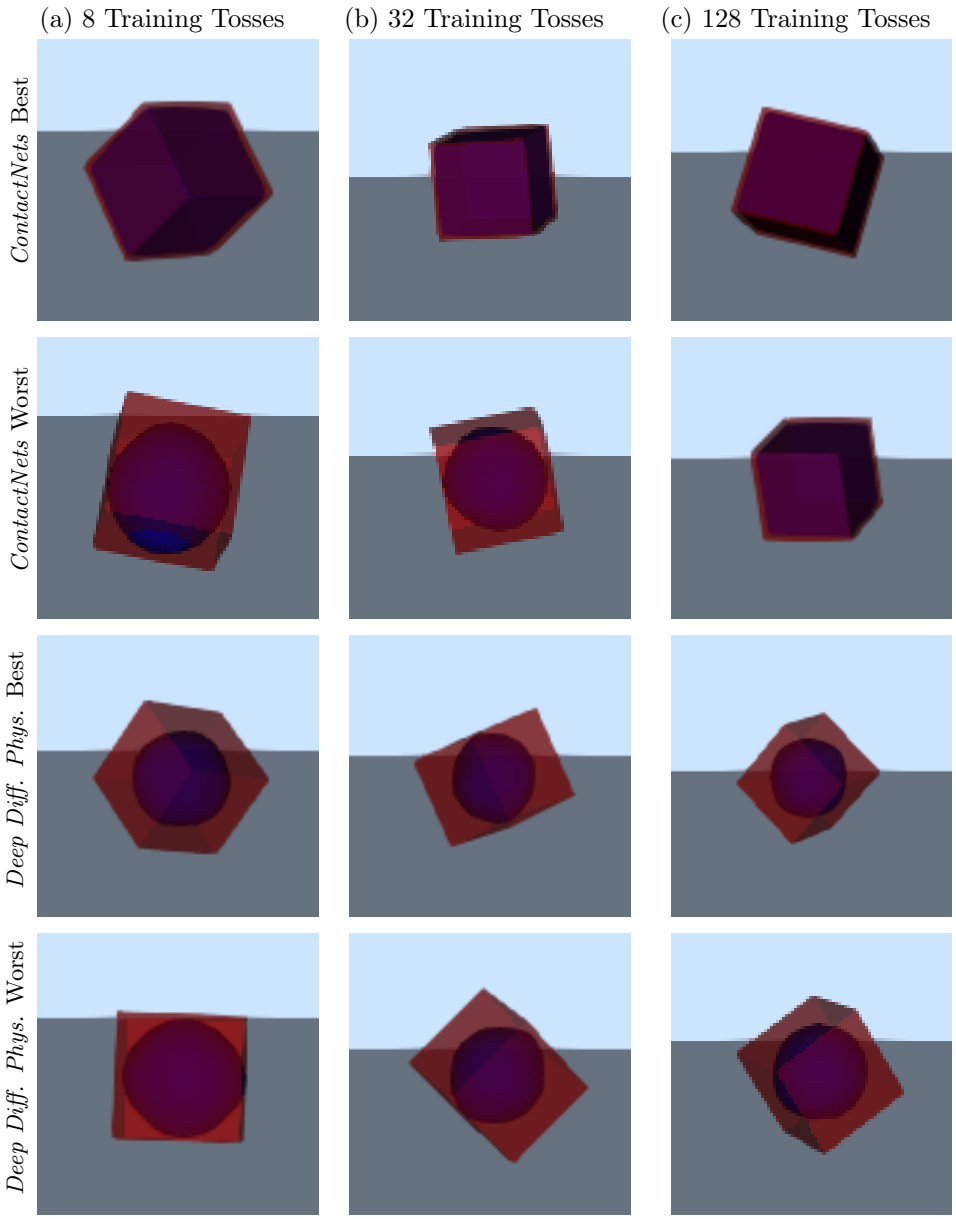


Figure 6.5: The ability of learned *ContactNets* and *Deep Differentiable Physics* models (both in blue blue) to recover ground-truth geometry (red) is compared. For each method and dataset size, the best- and worst-performance models are shown. Regardless of dataset set, *ContactNets*' best model recovers the ground-truth shape to high precision; at 128 tosses, all models converge to the true geometry. By contrast, *Deep Differentiable Physics* consistently, largely underestimates the cube's volume.

rigid-body models, we display snapshots of these trajectory rollouts, with learned (blue) and ground-truth (red) geometries visualized. For each method and dataset size, we show a snapshot from both the best- and worst- performing of 30 trials. *Deep Differentiable Physics* models always fails find the correct geometry, and instead find a round object inscribed in the true ground-truth shape. By contrast, even on just 8 training tosses, the best-performing *ContactNets* model recovers the ground truth geometry to high precision. Additionally, for 128 training tosses, the worst-performing model over 30 trials also recovers the geometry similarly well.

Across dataset sizes, both types of rigid-body models have excellent match between training and test set performance, implying that they both generalize well to unseen data, even when trained on just a few rollouts.

End-to-end models perform similar or worse than *ContactNets* on test data, despite superior performance in the low-data regime on training data. This means that these pure-DNN models, which lack physical structure, require a large amount of data to effectively achieve low generalization error.

6.5. Discussion

Discontinuous and non-unique impact and stiction underpin essential robotics tasks—thus capturing these phenomena in learned models is crucial for their effective use in the real world. Our method, *ContactNets*, presents a novel and theoretically-proven approach to resolving fundamental problems in representing these behaviors with neural networks, and produces realistic dynamics from limited training data. On a real-world example, we have demonstrated that both the novel representation and training objective embedded in *ContactNets* lead to improved accuracy of learned models across a wide range of dataset sizes.

The primary limitation of our model is the constrictive nature of its priors: namely that the analytical contact-free dynamics are known and exact; collisions are inelastic; and objects are rigid. Additionally, further experimentation involving a manipulator interacting with more complex objects would allow us to evaluate our formulation's ability to capture multi-body contact. To these ends, our current work (Bianchini, Halm, and Posa, 2023) is focused on integrating residual and inertial-

parameter learning into the *ContactNets* formulation, and application of the upgraded method to articulated objects thrown by a robot arm. Subsequent work may focus on verifing the quality of our learned models for executing robotic tasks by utilizing them in planning and control algorithms.

Additionally, as real-time data of object poses is unavailable in some applications, natural extensions could involve embedding our formulation into dynamical models based on visual data. Recent advances in keypoint-based approaches Manuelli et al. (2019) suggest a promising intermediate representation for inferring contact geometry from video. Additionally, point-cloud data from RGB-D sensors may be utilized for warm-starting or sensor fusion with our geometric representations, potentially enabling faster training convergence and higher-quality geometry estimation.

CHAPTER 7

Set-valued Rigid-body Dynamics for Simultaneous, Inelastic, Frictional Impacts

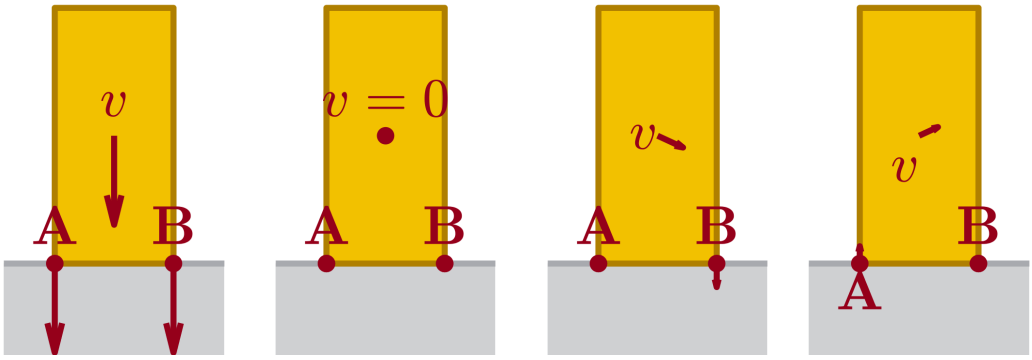
Parts of this chapter were previously published as parts of Mathew Halm and Michael Posa. Modeling and Analysis of Non-unique Behaviors in Multiple Frictional Impacts. In *Robotics: Science and Systems (RSS)*, Freiburg im Breisgau, Germany, 2019. URL http://roboticsproceedings.org/rss15/p22.pdf. The included portions represent original contributions.

7.1. Introduction

In Chapters 5 and 6, we focused on the process of recovering rigid-body model parameters from data. In this chapter, we consider that even if such estimated parameters are precisely identified, the rigid-body models built from them will never perfectly predict the outcome of every real-world scenario.

Real-world robotics systems must rely on state estimation method to approximately recover the system's current state. As discussed in Chapter 1, under the extreme sensitivity to initial conditions induced by contact, even slight errors in the state estimate may produce large prediction error. A familiar occurrence of this sensitivity is the unpredictability of billiards breaks and dice rolls, though even a single object impacting flat ground (Figure 7.1) is difficult to model. Unfortunately, sensitive, simultaneous impacts regularly occur in robotics (see Section 7.2.1).

This sensitivity is highly dependent the rapid ordering or sequencing of impact forces between the various colliding bodies, where are in turn extremely sensitive to the initial condition (Chatterjee, 1999; Hurmuzlu and Marghitu, 1994; Ivanov, 1995; Smith et al., 2012; Uchida et al., 2015; Wang et al., 1992). In reality, this ordering emerges from material properties and deformation dynamics (Chatterjee, 1999; Nguyen and Brogliato, 2018), which are generally not tractable to fully identify or simulate in real-world robotics scenarios. Instead, robotics methods rely on the rigid-body models discussed in this thesis, which inherently do not capture true impact physics; instead, typical models select a single outcome according to one of many heuristics, including maximum dissipa-



(a) Initial condition (b) Symmetric impact (c) A-then-B sequential impacts

Figure 7.1: (a) A rectangular block (yellow) with velocity v is dropped onto flat ground (gray), colliding at two corners; multiple physically-realistic results may emerge from these simultaneous impacts. (b) As the system is symmetric about the vertical axis, one possibility is that the phone comes to rest; the model in Anitescu and Potra (1997) produces this outcome. (c) With sufficient friction, any impact at a single corner can stick. If point A impacts first, the phone pivots counter-clockwise, causing a a second impact at B, which then causes the phone to pivot about B with A lifting off the ground. If the impacts are instead ordered B-then-A, then by symmetry is A pivots and B lifts off.

tion (Drumwright and Shell, 2010), minimum potential energy (Uchida et al., 2015), and symmetry (Smith et al., 2012). Other models (e.g. the LCP-based ones discussed in Section 2.5.4) have a handful of non-unique solutions, but rely on a numerical solvers which may be biased toward a particular solution (Anitescu and Potra, 1997; Stewart and Trinkle, 1996; Horak and Trinkle, 2019; Remy, 2017; Anitescu, 2005). Under restrictions on the systems and mechanics involved, such as massless limbs and no kinetic friction, such models may lead to useful, accurate modeling of robotic systems and tasks (Johnson et al., 2016; Burden et al., 2016). However, both simple and complex robotics systems violate these assumptions, and different heuristics inevitably produce disagreeing or unrealistic predictions (Remy, 2017; Fazeli et al., 2020b). Furthermore, their single-outcome nature does not reflect the large uncertainty generated from the practically-unknowable sequencing of impacts.

This chapter takes a fundamentally different perspective, in which we propose the development of *set-valued* rigid-body models that attempt to generate all physically-reasonable outcomes, by capturing all possible orders of impacts. Though some predictions from such a model may not

ultimately occur, controllers guaranteed to stabilize the model—and learned policies trained on the model's predictions—are well-positioned to perform reliably in the real world. While non-unique predictions through randomly-sequenced individual impacts has existed conceptually for decades (Ivanov, 1995), such methods do not capture the subtleties of partially-concurrent impacts, and feasible computation of the entire set of possibilities has remained an open problem (Stewart, 2000). In the domain of inelastic impacts, we tackle both of these issues by developing a differential impact model which allows impacts to resolve at arbitrary relative rates, first conceptually explored in Posa et al. (2016b). This construction is similar mathematically to other recent methods, including Darboux-Keller (Keller, 1986) and LZB (Nguyen and Brogliato, 2018) approaches, in that it extends Routh's original method for inelastic impact (Routh, 1891); such extensions have so far however been focused on producing a single outcome when well-identified material properties are available (Nguyen and Brogliato, 2018).

A prior version of this work was published in Halm and Posa (2019), provided the first detailed theoretical analysis of this construction, by encapsulating the resulting dynamics as a differential inclusion. This work supplements this work by embedding these impact dynamics into the continuous time evolution of rigid body systems, as well as a new numerical technique for tractable approximation of the set of outcomes to simultaneous impacts. The chapter is organized as follows:

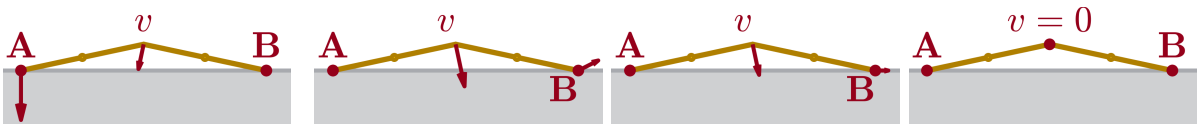
- In Section 7.2, we provide a simplified theoretical analysis of the set-valued impact model in Halm and Posa (2019), with proven existence of solutions (Lemma 3); energy dissipation (Lemma 5) impact termination (Theorem 6). We include new motivating examples highlighting the inconsistencies between existing models of simultaneous impact.

- In Section 7.3, we unify set-valued impacts and continuous-time evolution into a single model, with guaranteed existence of solutions (Theorem 7) and time advancement (Theorem 10).

- In Section 7.4, we formulate time-stepping simulation of the impact model as a linear complementarity problem (LCP). We provide bounds on computation time for both sampling from (Theorem 13) and global approximation of (Theorem 14) the feasible post-impact velocity set.

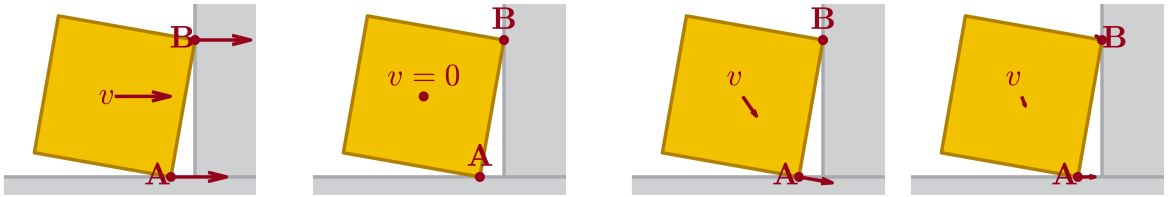
- In Section 7.5, we apply our model to several examples from robotic locomotion and manipulation.

7.2. Simultaneous Impact Model



(a) Initial condition (b) Simultaneous algebraic impacts

Figure 7.2: (a) A compass gait walker, consisting of two legs attached with a hinge joint at the hip, takes a step with hip velocity v and excites non-uniqueness in the model of Anitescu and Potra (1997). (b) A single impact at that the leading foot (point A) can cause the trailing foot (point B) to lift off the ground. Alternatively, impacts at both feet can cause the trailing foot to slide or come to rest.



(a) Initial condition (b) Simultaneous impact (c) B-then-A sequential impact

Figure 7.3: Subtly different solutions for a box sliding into a wall with velocity v (a) are shown. (b) When a simultaneous impact is generated via Anitescu and Potra (1997), the box comes to rest. (c) When point B has an impact before point A, point A instead continues sliding, while point B lifts off the wall.

While Figure 7.1 demonstrates that simultaneous collisions can excite meaningful disagreement between common impact models, making two points collide at *exactly* the same is unlikely in real life. However, as shown on a real-world system by Chatterjee (1999), even a single collision can result in multiple outcomes depending on the ordering of impulse accumulation between contacts. In this section, we first offer two additional examples of this type—one related to legged locomotion and the other to manipulation; technical details can be found in Appendix E.1. We then describe a model that captures the non-uniqueness due to impulse ordering by extending Routh's method to multiple contacts with arbitrary relative rates.

7.2.1. Motivating Examples

A ubiquitous model of bipedal walking is the compass gait walker, which consists of two rods (legs) connected with a revolute joint at the hip. Bipedal walking involves stepping with a leading foot while a trailing foot rests on the ground, as shown in Figure 7.2. As observed by Remy (2017), if a wide step (156° between the legs) is taken by the model, then the simultaneous method of Anitescu and Potra (1997) results in three categorically different solutions. In one case, there is only an impact at the leading foot, and the trailing foot lifts of the ground. In two others, impacts at both feet can result in the trailing foot sliding or coming to rest.

In the second example, motivated by non-prehensile pushing of an object, we consider a box which slides on one corner on a floor before impacting a wall (Figure 7.3). If a single impact occurs between the box and the wall, it will trigger a second impact against the floor. Due to the position of the center of mass of the box, both impacts add counter-clockwise rotational momentum to the box, causing the contact with the wall to lift off. Alternatively, if both of these impacts are resolved simultaneously, the box comes to rest under sufficient friction.

7.2.2. Simultaneous Impact Model Construction

We have previously demonstrated that some simultaneous impact models are sensitive to impulse ordering. As predicting this ordering demands precise knowledge of initial conditions and material properties beyond the fidelity of robotic sensors and simplified rigid-body models, we instead seek to predict the set of outcomes that result from arbitrary impulse orders.

The foundational concept of this model is that while Routh's method models impacts as instantaneous (Routh, 1891), the variable of integration s provides a natural way to specify the relative rates of impulse accural between concurrent impacts. A similar model, without theoretical results or a detailed understanding, was proposed by Posa et al. (2016b) where it proved useful for stability analysis of robots undergoing simultaneous impact. We consider the following extension to Routh's method which at any given instant during the resolution process, the impacts are allowed to concurrently resolve at *any* relative rate:

1. Increase $\boldsymbol{\Lambda}_{n,i}$ on each non-separating ($\boldsymbol{J}_{n,i}\boldsymbol{v} \leq 0$) active contact $i \in \mathcal{I}_A(\boldsymbol{q})$ at rate $\boldsymbol{\lambda}_{n,i} \geq 0$ such that
$$\sum_i \boldsymbol{\lambda}_{n,i} = \|\boldsymbol{\lambda}_n\|_1 = 1\,. \tag{7.1}$$

2. Increment each tangential impulse with slope $\boldsymbol{\lambda}_{t,i}$ satisfying Coulomb friction (2.14) at $\bar{\boldsymbol{v}} = \boldsymbol{v} + \boldsymbol{M}^{-1}\boldsymbol{J}^T\boldsymbol{\Lambda}$.

3. Terminate when all $\boldsymbol{J}_{n,i}\bar{\boldsymbol{v}} \geq 0$, i.e. $\bar{\boldsymbol{v}} \notin \mathcal{C}(\boldsymbol{q})$. $\boldsymbol{v}^+ = \bar{\boldsymbol{v}}$.

We can understand the constraint (7.1) on $\boldsymbol{\lambda}$ as choosing a net force that comes from a convex combination of the forces that Routh's method might select for any of the individual contacts $i \in \mathcal{I}_A(\boldsymbol{q})$. In particular, we note that step 1 restricts the normal forces to be dissipative, i.e.

$$\boldsymbol{\lambda}_{n,i} \cdot \boldsymbol{J}_{n,i}\boldsymbol{v} \leq 0\,. \tag{7.2}$$

As before, we can capture this behavior as a DI:

$$\dot{\boldsymbol{v}} \in D_{\boldsymbol{q}}(\boldsymbol{v}) = \boldsymbol{M}^{-1}\mathrm{co}\left(\bigcup_{i\in\mathcal{I}_{\boldsymbol{q}}(\boldsymbol{v})} F_i(\boldsymbol{q},\boldsymbol{v},1)\right)\,, \tag{7.3}$$

$$\mathcal{I}_{\boldsymbol{q}}(\boldsymbol{v}) = \begin{cases} \{i \in \mathcal{I}_A(\boldsymbol{q}) : \boldsymbol{J}_{n,i}\boldsymbol{v} \leq 0\} & \boldsymbol{v} \in \mathrm{cl}\mathcal{C}(\boldsymbol{q})\,, \\ \arg\min_{i\in\mathcal{I}_A(\boldsymbol{q})} \boldsymbol{J}_{n,i}\boldsymbol{v} & \text{otherwise}\,. \end{cases} \tag{7.4}$$

While non-physical, the behavior outside of $\mathrm{cl}\mathcal{C}(\boldsymbol{q})$ has been chosen to preserve upper semi-continuity, and is not encountered when resolving impacts due to the termination condition $\boldsymbol{v} \notin \mathcal{C}(\boldsymbol{q})$. The construction of (7.3) is similar to that of the single contact system (2.34); it is furthermore equivalent to (2.34) and therefore Routh's method when only one contact is active.

7.2.3. Properties

We now detail properties of our multi-contact impact system that are useful for analyzing its solution set.

Existence and Closure

For any configuration $\boldsymbol{q} \in \mathcal{Q}_A$, $D_{\boldsymbol{q}}(\boldsymbol{v})$ is non-empty, closed, uniformly bounded, and convex. Therefore by Proposition 2, we obtain the following:

Lemma 3 (Existence of Solutions (Appendix E.3.1))**.** *For all configurations $\boldsymbol{q} \in \mathcal{Q}_A$, velocities $\boldsymbol{v}_0$, and compact intervals $[a, b]$,* $\mathrm{SOL}\left(D_{\boldsymbol{q}}, [a, b]\right)$ *and* $\mathrm{IVP}\left(D_{\boldsymbol{q}}, \boldsymbol{v}_0, [a, b]\right)$ *are non-empty and closed under uniform convergence.*

Homogeneity

As the set of allowable contact forces are only dependent on the direction of $\boldsymbol{v}$, $D_{\boldsymbol{q}}(\boldsymbol{v})$ is positively homogeneous in $\boldsymbol{v}$, i.e., $\forall k > 0, \boldsymbol{v} \in \mathbb{R}^{n_v}$, $D_{\boldsymbol{q}}(\boldsymbol{v}) = D_{\boldsymbol{q}}(k\boldsymbol{v})$. Positive homogeneity induces a similar property on the solution set to the differential inclusion:

Lemma 4 (Homogeneity (Appendix E.3.2))**.** *For all $\boldsymbol{q}$, $k > 0$, and $[a, b]$ compact, if $\boldsymbol{v}(s) \in \mathrm{SOL}\left(D_{\boldsymbol{q}}, [a, b]\right)$, $k\boldsymbol{v}(\frac{s}{k}) \in \mathrm{SOL}\left(D_{\boldsymbol{q}}, [ka, kb]\right)$.*

Energy Dissipation

An essential behavior of inelastic impacts reflected in our model is that they dissipate kinetic energy. By construction of (7.3), the kinetic energy $K(\boldsymbol{q}, \boldsymbol{v}(s))$ is continually non-increasing during impact (i.e. when $\boldsymbol{v}(s) \in \mathrm{cl}\mathcal{C}(\boldsymbol{q})$) as normal forces are constrained to be dissipative (7.2) and frictional forces are naturally, maximally dissipative:

Lemma 5 (Dissipation (Appendix E.3.3))**.** *Let $\boldsymbol{q} \in \mathcal{Q}_A$, and let $[a, b]$ be a compact interval. If $\boldsymbol{v}(s) \in \mathrm{SOL}\left(D_{\boldsymbol{q}}, [a, b]\right)$ and $\boldsymbol{v}([a, b]) \subseteq \mathrm{cl}\mathcal{C}(\boldsymbol{q})$, then $\|\boldsymbol{v}(s)\|_{\boldsymbol{M}}$ is non-increasing.*

K has total derivative

$$\dot{K} = \boldsymbol{v}^T \boldsymbol{J}^T \boldsymbol{\lambda}\,, \tag{7.5}$$

and it will strictly decrease unless the velocity is constant:

Theorem 4 (Strong Dissipation (Appendix E.3.4))**.** *Let $\boldsymbol{q} \in \mathcal{Q}_A$, and let $[a, b]$ be compact. If $\boldsymbol{v}(s) \in \mathrm{SOL}\left(D_{\boldsymbol{q}}, [a, b]\right)$ and $\boldsymbol{v}\left([a, b]\right) \subseteq \mathrm{cl}\mathcal{C}(\boldsymbol{q})$, $\|\boldsymbol{v}(s)\|_{\boldsymbol{M}}$ constant implies $\boldsymbol{v}(s)$ constant.*

One might then wonder if K strictly decreases during impact; certainly, this would not be the case if $\boldsymbol{v}(s)$ could stay constant. Therefore, solutions to the differential inclusion must not be permitted to select $\dot{\boldsymbol{v}} = \mathbf{0}$, i.e., $\mathbf{0} \notin D_{\boldsymbol{q}}(\boldsymbol{v})$ for every $\boldsymbol{v} \in \mathrm{cl}\mathcal{C}(\boldsymbol{q})$; As $D_{\boldsymbol{q}}(\boldsymbol{v}) \subseteq \boldsymbol{M}^{-1}\mathrm{FC}(\boldsymbol{q})$, this property is guaranteed by the pointed friction cone assumption (Assumption 2). Assumption 2 covers most situations in robotics—including grasping and locomotion—with the notable exception being jamming between immovable surfaces. Furthermore, it guarantees strict dissipation during the impact process:

Theorem 5 (Strict Dissipation)**.** *Let $\boldsymbol{q} \in \mathcal{Q}_A \setminus \mathcal{Q}_P$ and $[a, b]$ be a compact interval. If $\boldsymbol{v}(s) \in \mathrm{SOL}\left(D_{\boldsymbol{q}}, [a, b]\right)$ and $\boldsymbol{v}([a, b]) \subseteq \mathrm{cl}\mathcal{C}(\boldsymbol{q})$, $\|\boldsymbol{v}(s)\|_{\boldsymbol{M}}$ is strictly decreasing.*

Proof. As $\boldsymbol{v}$ is never constant on $\mathrm{cl}\mathcal{C}(\boldsymbol{q})$ via Assumption 2, $\|\boldsymbol{v}(s)\|_{\boldsymbol{M}}$ strictly decreases by Theorem 4 and Lemma 5. □

7.2.4. Linear Impact Termination

While solutions to the underlying DI are guaranteed to exist in the simultaneous impact model, we have yet to prove that they terminate the impact process, as in Routh's single-contact method. We now discuss a similar linear-duration condition:

Proposition 6 (Finite Termination)**.** *For any configuration $\boldsymbol{q} \in \mathcal{Q}_A \setminus \mathcal{Q}_P$ and pre-impact velocity $\boldsymbol{v}(0)$, the DI (7.3) resolves the impact within a duration proportional to $\|\boldsymbol{v}(0)\|_{\boldsymbol{M}}$.*

We will prove this claim as a consequence of kinetic energy decreasing fast enough to force termination, a significant expansion of Theorem 5. Even though K always decreases, Theorem 5 does not forbid $\dot{K}$ from getting arbitrarily close to zero. For example, consider a 2 DoF system with 2 frictionless, axis-aligned contacts ($\boldsymbol{M} = \boldsymbol{J}_n = \boldsymbol{I}_2$). For any $\epsilon > 0$, we can pick a velocity and impulse increment which satisfy $\dot{K} > -\epsilon$:

$$\boldsymbol{v}_\epsilon = \frac{-1-\epsilon}{2}\begin{bmatrix}1\\ \epsilon\end{bmatrix} \in \mathcal{C}(\boldsymbol{q})\,, \qquad \boldsymbol{J}_n^T\begin{bmatrix}\epsilon\\ 1\end{bmatrix}\frac{1}{1+\epsilon} \in D_{\boldsymbol{q}}(\boldsymbol{v}_\epsilon)\,. \tag{7.6}$$

However as we take $\epsilon \to 0$, $\boldsymbol{v}_\epsilon$ converges to a non-impacting velocity; thus, $\dot{K}$ only remains small for a short duration before impact termination. It remains possible that the aggregate dissipation over an interval of nonzero length can be bounded away from zero. We define this quality as $\alpha(s)$-dissipativity:

Definition 6 ($\alpha(s)$-dissipativity)**.** *For a positive definite function $\alpha(s) : \mathrm{cl}\mathbb{R}^+ \to [0,1]$, the system $\dot{\boldsymbol{v}} \in D_{\boldsymbol{q}}(\boldsymbol{v})$ is said to be $\alpha(s)$-**dissipative** if for all $s > 0$, for all $\boldsymbol{v} \in \mathrm{SOL}\left(D_{\boldsymbol{q}}, [0,s]\right)$ s.t. $\boldsymbol{v}\left([0,s]\right) \subseteq \mathrm{cl}\mathcal{C}(\boldsymbol{q})$, if $\|\boldsymbol{v}(0)\|_{\boldsymbol{M}} = 1$, $\|v(s)\|_{\boldsymbol{M}} \leq 1 - \alpha(s)$.*

$\alpha(s)$-dissipativity can be understood as a homogeneous dissipation rate. It therefore induces the desired linear-duration termination in Proposition 6:

Lemma 6 (Termination via Aggregate Dissipation (Appendix E.3.5))**.** *Let $\boldsymbol{q} \in \mathcal{Q}_A$ and let $\dot{\boldsymbol{v}} \in D_{\boldsymbol{q}}(\boldsymbol{v})$ be $\alpha_{\boldsymbol{q}}(s)$-dissipative. Then if $\boldsymbol{v}(s) \in \mathrm{SOL}\left(D_{\boldsymbol{q}}, [0,s^*]\right)$ and $\boldsymbol{v}([0,s^*]) \subseteq \mathrm{cl}\mathcal{C}(\boldsymbol{q})$,*

$$s^* \leq \inf_{s>0} \frac{s\,\|\boldsymbol{v}(0)\|_{\boldsymbol{M}}}{\alpha_{\boldsymbol{q}}(s)}.$$

Under Assumption 2, *every* $\boldsymbol{q} \in \mathcal{Q}_A \setminus \mathcal{Q}_P$ exhibits $\alpha(s)$-dissipativity, a direct proof of Proposition 6:

Theorem 6 (Aggregate Dissipation (Appendix E.3.6))**.** *For every configuration $\boldsymbol{q} \in \mathcal{Q}_A \setminus \mathcal{Q}_P$ there exists an $\alpha_{\boldsymbol{q}}(s)$ such that $\dot{\boldsymbol{v}} \in D_{\boldsymbol{q}}(\boldsymbol{v})$ is $\alpha_{\boldsymbol{q}}(s)$-dissipative.*

The u.s.c. structure of $D_{\boldsymbol{q}}$ has the additional implication that nearby configurations obey a uniform dissipation rate:

Corollary 1 (Uniform Aggregate Dissipation (Appendix E.3.7))**.** *For compact $\mathcal{Q} \subseteq \mathcal{Q}_A \setminus \mathcal{Q}_P$, there exists a single $\alpha_{\mathcal{Q}}(s)$ such that $\dot{\boldsymbol{v}} \in D_{\boldsymbol{q}}(\boldsymbol{v})$ is $\alpha_{\mathcal{Q}}(s)$-dissipative for all $\boldsymbol{q} \in \mathcal{Q}$.*

7.3. Continuous-Time Dynamics Model

We now describe how the simultaneous impact DI can be embedded into a full, continuous-time dynamics model. As the impact model integrates over a variable other than time, rather than

switching between integration spaces, we define time advancement t as a variable in an augmented state $\bar{\boldsymbol{x}}(s)$:

$$\bar{\boldsymbol{x}}(s) = \begin{bmatrix} \boldsymbol{x}(s) \\ t(s) \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}(s) \\ \boldsymbol{v}(s) \\ t(s) \end{bmatrix} \in \mathbb{R}^{n_q+n_v+1} . \tag{7.7}$$

For any state $\bar{\boldsymbol{x}}(s)$ we can extract the relevant configuration, velocity, and time as by selecting the appropriate indices, e.g. as $\boldsymbol{q}(\bar{\boldsymbol{x}}(s))$. For notational compactness, whenever clear, we will write this construction in the shortened form $\boldsymbol{q}(s)$. We will also frequently make use of the sets

$$\bar{\mathcal{X}}_A = \{\bar{\boldsymbol{x}} : \boldsymbol{q}(\bar{\boldsymbol{x}}) \in \mathcal{Q}_A\} , \quad \bar{\mathcal{X}}_P = \{\bar{\boldsymbol{x}} : \boldsymbol{q}(\bar{\boldsymbol{x}}) \in \mathcal{Q}_P\} . \tag{7.8}$$

7.3.1. Model Construction

We now construct the dynamics model as a differential inclusion $\frac{\mathrm{d}}{\mathrm{d}s}\bar{\boldsymbol{x}}(s) \in D(\bar{\boldsymbol{x}}(s))$. Under this formulation, the velocity $\boldsymbol{v}(s)$ is continuous with respect to s, but can be discontinuous with respect to time $t(s)$ in the sense that $\boldsymbol{v}$ can evolve while t is held constant. To make the system autonomous, we represent the external forces $\boldsymbol{u}$ as set-valued, time-varying full-state feedback $\mathcal{U}(\bar{\boldsymbol{x}})$. In order for the system to be well-behaved, we assume that the convex-compact u.s.c. properties exploited in the impact dynamics carry over into the continuous time case, because $\mathcal{U}$ itself has these properties:

Assumption 4. *$F_s(\boldsymbol{x},\mathcal{U}(\bar{\boldsymbol{x}}))$ is convex-compact u.s.c. in $\bar{\boldsymbol{x}}$.*

We identify three behaviors that $\dot{\bar{\boldsymbol{x}}} \in D(\bar{\boldsymbol{x}})$ should obey:

No Contact Forces

Whenever all active contacts have separating velocities (and when no contacts are active), i.e.

$$\bar{\boldsymbol{x}}(s) \in \bar{\mathcal{X}}_{\mathcal{S}} = \{\bar{\boldsymbol{x}} : \boldsymbol{v}(\bar{\boldsymbol{x}}) \in \mathcal{S}(\boldsymbol{q}(\bar{\boldsymbol{x}}))\} , \tag{7.9}$$

$\bar{\boldsymbol{x}}(s)$ should evolve according to (2.8) with no contact forces ($\boldsymbol{\lambda} = \mathbf{0}$), in the sense that

$$\boldsymbol{M}(\boldsymbol{q})\mathrm{d}\boldsymbol{v} \in F_s\left(\boldsymbol{x}, \mathcal{U}(\bar{\boldsymbol{x}})\right)\mathrm{d}s\,, \qquad\qquad \mathrm{d}s = \mathrm{d}t\,. \tag{7.10}$$

these equations can be packaged into DI form as

$$\dot{\bar{\boldsymbol{x}}} \in D_{\mathcal{S}}(\bar{\boldsymbol{x}}) = \begin{bmatrix} \boldsymbol{\Gamma}\boldsymbol{v} \\ \boldsymbol{M}^{-1}F_s(\boldsymbol{x}, \mathcal{U}(\bar{\boldsymbol{x}})) \\ 1 \end{bmatrix}. \tag{7.11}$$

Collision

Whenever $\boldsymbol{v}(s)$ is colliding over $[a, b]$, i.e.

$$\bar{\boldsymbol{x}}([a,b]) \subseteq \bar{\mathcal{X}}_{\mathcal{C}} = \{\bar{\boldsymbol{x}} : \boldsymbol{v}(\bar{\boldsymbol{x}}) \in \mathcal{C}(\boldsymbol{q}(\bar{\boldsymbol{x}}))\}\,, \tag{7.12}$$

t and $\boldsymbol{q}$ should be constant, and $\boldsymbol{v}$ should obey our simultaneous impact model:

$$\dot{\bar{\boldsymbol{x}}} \in D_{\mathcal{C}}(\bar{\boldsymbol{x}}) = \begin{bmatrix} \mathbf{0} \\ D_{\boldsymbol{q}}(\boldsymbol{v}) \\ 0 \end{bmatrix}. \tag{7.13}$$

Sustained Contact

The model must capture continuous state evolution with respect to time under sustained contact, as in (2.8). Additionally, proving that our model is well-behaved requires that $D(\bar{\boldsymbol{x}}(s))$ be convex. Conveniently, sustained contact can be represented as a convex combination of contactless and collision dynamics:

$$\dot{\bar{\boldsymbol{x}}}(s) \in \mathrm{co}\left(D_{\mathcal{S}}(\bar{\boldsymbol{x}}) \cup D_{\mathcal{C}}(\bar{\boldsymbol{x}})\right)\,. \tag{7.14}$$

To demonstrate this property, we consider that (2.8) dictates that the state $\boldsymbol{q}$, $\boldsymbol{v}$ under sustained contact obeys

$$\mathrm{d}\boldsymbol{q} = \boldsymbol{\Gamma}\boldsymbol{v}\mathrm{d}t\,, \qquad\qquad \boldsymbol{M}\mathrm{d}\boldsymbol{v} \in \left(\boldsymbol{J}^T\boldsymbol{\lambda} + F_s\right)\mathrm{d}t\,, \tag{7.15}$$

for finite, non-zero contact forces $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_n;\ \boldsymbol{\lambda}_t]$. Letting $\tilde{\boldsymbol{\lambda}} = \frac{\boldsymbol{\lambda}}{\|\boldsymbol{\lambda}_n\|_1}$, our impact model would allow $\boldsymbol{M}\mathrm{d}\boldsymbol{v} \in \boldsymbol{J}^T\tilde{\boldsymbol{\lambda}}\mathrm{d}s$ at $\boldsymbol{q}, \boldsymbol{v}$. Thus selecting $\dot{t} = \frac{1}{1+\|\boldsymbol{\lambda}_n\|_1} \in (0,1)$, we rewrite (7.15) as

$$\mathrm{d}\boldsymbol{q} = ((1-\dot{t})\boldsymbol{0} + \dot{t}\boldsymbol{\Gamma}\boldsymbol{v})\mathrm{d}s\,, \tag{7.16}$$

$$\boldsymbol{M}\mathrm{d}\boldsymbol{v} \in \left((1-\dot{t})\boldsymbol{J}^T\tilde{\boldsymbol{\lambda}} + \dot{t}F_s(\boldsymbol{x}, \mathcal{U}(\bar{\boldsymbol{x}}))\right)\mathrm{d}s\,, \tag{7.17}$$

$$\mathrm{d}t = ((1-\dot{t})0 + \dot{t}1)\mathrm{d}s\,. \tag{7.18}$$

The convex combination DI (7.14) can then generate sustained contact with this choice of $\dot{t}$. As a result, $t(s)$ neither evolves directly with s nor remains constant; effectively, solutions of (7.14) slow down time by a factor of $(1 + \|\boldsymbol{\lambda}_n\|_1)$. We will show that this factor is bounded on average under mild assumptions.

We now combine these three modes into a single differential inclusion. While we might easily chose the contactless mode when $\bar{\boldsymbol{x}} \in \bar{\mathcal{X}}_{\mathcal{S}}$, arbitrating between impact and sustained contact when the velocity is non-separating is less obvious, particularly as Painlevé's Paradox (see Stewart (2000) for details) might require impact dynamics even without a collision. Furthermore, almost all selections of $\dot{\bar{\boldsymbol{x}}}$ from $\mathrm{co}\left(D_{\mathcal{S}}(\bar{\boldsymbol{x}}) \cup D_{\mathcal{C}}(\bar{\boldsymbol{x}})\right)$ will correspond to non-physical behavior; a particular $\dot{\bar{\boldsymbol{x}}}$ must be chosen to *maintain* contact by exactly counteracting forces such that inter-body distance is *identically* zero during contact.

We resolve these issues in an implicitly in the full DI

$$\dot{\bar{\boldsymbol{x}}} \in D(\bar{\boldsymbol{x}}) = \begin{cases} D_{\mathcal{S}}(\bar{\boldsymbol{x}}) & \bar{\boldsymbol{x}} \in \bar{\mathcal{X}}_{\mathcal{S}}\,, \\ D_{\mathcal{C}}(\bar{\boldsymbol{x}}) & \bar{\boldsymbol{x}} \in \operatorname{int}(\bar{\mathcal{X}}_{\mathcal{C}})\,, \\ \operatorname{co}\left(D_{\mathcal{S}}(\bar{\boldsymbol{x}}) \cup D_{\mathcal{C}}(\bar{\boldsymbol{x}})\right) & \text{otherwise}\,. \end{cases} \tag{7.19}$$

We will show that in this model, $\boldsymbol{\phi}(\boldsymbol{q}) = \mathbf{0}$ is effectively a *barrier*: solutions beginning at a non-penetrating configuration are forced to *never* penetrate.

7.3.2. Properties

Existence and Closure

As we previously reviewed, existence guarantees for continuous-time evolution through impact have thus far been severely limited. We now show that our philosophy of including a wide set of behaviors leads to existence of solutions via Proposition 2, and the only additional assumptions required are that energy and inputs are bounded (Assumptions 5 and 6). The continuous-time DI (7.19) directly exhibits many of the properties required for Proposition 2. B its construction, at any $\bar{\boldsymbol{x}}$, $D(\bar{\boldsymbol{x}})$ is non-empty, compact, and convex. We will additionally see that it is u.s.c. in our proof of Theorem 7. However, as Coriolis components of F_s can grow quadratically, D is often not uniformly bounded; thus Proposition 2 cannot be directly used to prove existence of solutions. However, nearly identical properties of IVP's can still be established in the following manner. Suppose first that smooth forces can only input power at a bounded rate:

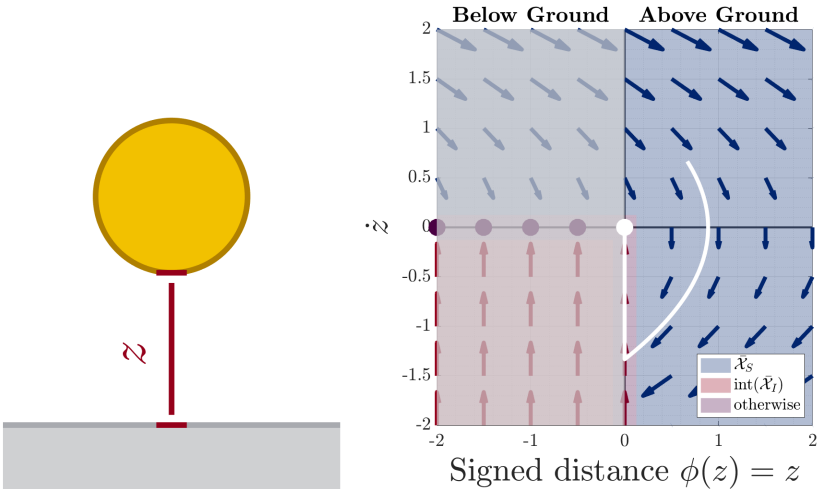
Assumption 5. $\exists c > 0$, $\boldsymbol{v} \cdot F_s(\boldsymbol{x}, \mathcal{U}(\bar{\boldsymbol{x}})) \leq c \left\|\boldsymbol{v}\right\|_{\boldsymbol{M}}$.

This condition is widely satisfied by many robotic systems, including those with globally bounded controllers and potential gradients (such as gravity). Assumption 5 implies that $\bar{\boldsymbol{x}}$ cannot diverge to infinity over a finite horizon. Furthermore, we will assume that if $\bar{\boldsymbol{x}}$ is bounded, $\dot{\bar{\boldsymbol{x}}}$ is bounded as well:

Assumption 6. *Over any compact set $\bar{\mathcal{X}}$, $F_s(\boldsymbol{x}, \mathcal{U}(\bar{\boldsymbol{x}}))$ is bounded, and therefore $D(\bar{\mathcal{X}})$ is compact.*

Assumptions 5 and 6 imply that over a finite interval, the solutions $\bar{\boldsymbol{x}}(s)$ beginning from a compact set $\bar{\mathcal{X}}$ have bounded derivative and therefore inherit the key existence, closure, and u.s.c. structure of globally bounded DI's:

Theorem 7 (Existence of Solutions (Appendix E.4.1))**.** *Let $\bar{\mathcal{X}}$ be a compact set and $[a, b]$ be a compact interval. Then* IVP $\left(D, \bar{\mathcal{X}}, [a, b]\right)$ *is compact and* IVP $(D, \bar{\boldsymbol{x}}, [a, b])$ *is non-empty, closed, convex, and u.s.c. in $\bar{\boldsymbol{x}}$ over $\bar{\mathcal{X}}$.*



(a) 1D ball-ground system (b) 1D system phase portrait

Figure 7.4: (a) A simple, 1D system of a non-rotating ball falling under gravity with configuration $\boldsymbol{q} = z = \boldsymbol{\phi}(z)$ is shown. (b) A phase plot demonstrates why our DI prevents penetration; an example trajectory is shown in white. Penetration corresponds to crossing from the right-half- to the left-half-plane. This cannot happen on the top half of the vertical axis ($\boldsymbol{\phi} = 0$, $\dot{z} \geq 0$), as the flow by definition points right. The cross also cannot happen on the bottom half of the axis, as the quadrant III has purely-vertical flow ($\mathrm{d}\boldsymbol{\phi} = \dot{z}\mathrm{d}t = 0$).

Non-Penetration

While there is no structure in $D(\bar{\boldsymbol{x}})$ that explicitly prevents penetration, $\boldsymbol{\phi}(\boldsymbol{q}) \geq \mathbf{0}$ is naturally, implicitly preserved, as the DI requires $\boldsymbol{\phi}_i(\boldsymbol{q}(s))$ to be constant under penetration. A graphical argument is given in Figure 7.4.

Theorem 8 (Non-Penetration (Appendix E.4.2))**.** *Let $\bar{\boldsymbol{x}}_0 \notin \bar{\mathcal{X}}_P$ be non-penetrating, let $[a, b]$ be a compact interval, and let $\bar{\boldsymbol{x}}(s) \in$* IVP $(D, \bar{\boldsymbol{x}}_0, [a, b])$*. Then $\bar{\boldsymbol{x}}(s) \notin \bar{\mathcal{X}}_P$ for all $s \in [a, b]$.*

Correct Mode Selection

Our requirements dictate that solutions $\bar{\boldsymbol{x}}(s)$ containing only separating velocities ($\bar{\boldsymbol{x}} \in \bar{\mathcal{X}}_{\mathcal{S}}$) should comply with contactless dynamics, and likewise with impact dynamics when $\bar{\boldsymbol{x}}(s)$ contains only impacting velocities and non-penetrating configurations ($\bar{\boldsymbol{x}} \in \bar{\mathcal{X}}_{\mathcal{C}} \setminus \bar{\mathcal{X}}_P$). The former is a trivial result of the construction of D, but the latter is only similarly trivial when $\bar{\boldsymbol{x}} \in \text{int}(\bar{\mathcal{X}}_{\mathcal{C}}) \setminus \bar{\mathcal{X}}_P$. However, all states $\bar{\boldsymbol{x}} \subseteq \bar{\mathcal{X}}_{\mathcal{C}}$ have penetrating velocity, and thus any contactless dynamics component in $\dot{\bar{\boldsymbol{x}}}$ would by definition cause $\bar{\boldsymbol{x}}$ to penetrate (i.e. enter $\bar{\mathcal{X}}_P$), allowing a proof by contradiction:

Theorem 9 (Impact Dynamics (Appendix E.4.3))**.** *Let $[a, b]$ be a compact interval and $\bar{\boldsymbol{x}}(s) \in \text{SOL}\,(D, [a, b])$ with $\bar{\boldsymbol{x}}([a, b]) \subseteq \bar{\mathcal{X}}_{\mathcal{C}} \setminus \bar{\mathcal{X}}_P$. Then $\bar{\boldsymbol{x}}(s) \in \text{SOL}\,(D_{\mathcal{C}}, [a, b])$.*

7.3.3. Linear Time Advancement

While Theorem 7 guarantees existence of solutions over any interval of s, practical application often requires reasoning about solution sets over intervals in time (over $t(s)$). To do so, solutions of the model must significantly advance time—i.e. for any time duration t_f, all solutions of the model have $t(s_f) - t(0) > t_f$ for large enough s_f. For small enough t_f, this property only requires the solution to exit the impact dynamics regime, which by Theorem 6 is guaranteed to occur:

Theorem 10 (Time Advancement (Appendix E.4.4))**.** *Let $\bar{\mathcal{X}} \subseteq \bar{\mathcal{X}}_P^c$ be a compact set with no penetrating configurations. Then there exists $s^*(\bar{\mathcal{X}}), t^*(\bar{\mathcal{X}}) > 0$, such that for all $s_f > s^*(\bar{\mathcal{X}})$, if $\bar{\boldsymbol{x}}(s) \in \text{IVP}\left(D, \bar{\mathcal{X}}, [0, s_f]\right)$, then $t(s_f) - t(0) > t^*(\bar{\mathcal{X}})$.*

If $t(s_f) - t(0) > t^*$ is guaranteed over a set $\bar{\mathcal{X}}$, then $t(s)$ must at least advance at rate $\frac{t^*}{s^*}$ over arbitrarily long horizons:

Corollary 2 (Amortized Advancement (Appendix E.4.5))**.** *Let $\bar{\mathcal{X}} \subseteq \bar{\mathcal{X}}_P^c$ be a compact set with no penetrating configurations, such that*

$$\bar{\mathcal{X}}(s_f) = \left\{\bar{\boldsymbol{x}}(\cdot) \in \text{IVP}\left(D, \bar{\mathcal{X}}, [0, s_f]\right) : \bar{\boldsymbol{x}}([0, s_f]) \subseteq \bar{\mathcal{X}}\right\}, \tag{7.20}$$

is non-empty for all $s_f > 0$. *Define* $s^*(\bar{\mathcal{X}}), t^*(\bar{\mathcal{X}}) > 0$ *as in Theorem 10, and let*

$$t_f(s_f) = \min_{\bar{\boldsymbol{x}}(\cdot) \in \bar{\mathcal{X}}(s_f)} t(\bar{\boldsymbol{x}}(s_f)) - t(\bar{\boldsymbol{x}}(0)) \,. \tag{7.21}$$

Then, $\liminf_{s_f \to \infty} \frac{t_f(s_f)}{s_f} \geq \frac{t^*(\bar{\mathcal{X}})}{s^*(\bar{\mathcal{X}})}$.

7.4. Discrete Impact Integration

Section 7.2 provides a rigorous theoretical framework guaranteeing existence of solutions to our impact model. In this section, we now develop a computational framework that allows this model to be applied to two key settings. In the first, we consider embedding this model into an "event-based" discrete-time simulation environment, such as the one developed in Anitescu and Potra (1997), where collisions are resolved instantaneously via an update function

$$\boldsymbol{v}^+ \leftarrow \text{ImpactLaw}(\boldsymbol{q}, \boldsymbol{v}) \,. \tag{7.22}$$

Faithful capture of the non-uniqueness in our model means that ImpactLaw should be capable of returning a set of different values for $\boldsymbol{v}^+$; simulation will be considered as sampling stochastically from this set. Our second application, reachability analysis, is to approximate the entire set of possible outcomes for a given initial condition. In pursuit of both applications, we develop an LCP-based integration scheme for our impact DI (7.3). We will bound the number of LCP solves required for each of these applications.

7.4.1. Model Construction

Just as forward Euler integration can cause penetration in continuous-time simulators (**?**), it can also break the inelastic condition if applied to our impact model (7.3). To rectify this issue, we develop an approximate, implicit, and discrete integration scheme. Our method takes a simulation

step by finding a small contact impulse increment $\boldsymbol{\lambda}$, such that

$$\text{find} \qquad \boldsymbol{v}'; \{\boldsymbol{\lambda}_i : i \in \mathcal{I}_A\} \;, \tag{7.23a}$$

$$\text{s.t.} \qquad \text{impulse balance (2.28)} \;, \tag{7.23b}$$

$$\boldsymbol{\lambda}_{n,i} \geq \mathbf{0} \text{ and } \boldsymbol{\lambda}_{n,i} \boldsymbol{J}_{n,i} \boldsymbol{v}' \leq 0 \,, \tag{7.23c}$$

$$\boldsymbol{\lambda}_{t,i} \in -\boldsymbol{\mu}_i \boldsymbol{\lambda}_{n,i} \text{Unit}_{\boldsymbol{D}}(\boldsymbol{J}_{t,i} \boldsymbol{v}') \,. \tag{7.23d}$$

where the dependence of $\boldsymbol{J}$ on $\boldsymbol{q} = \boldsymbol{q}_0$ is suppressed.

Conceptually, our model can be understood as differential, as it closely mirrors our impact DI (7.3) which selects $\dot{\boldsymbol{v}} \in \boldsymbol{M}^{-1}\text{FC}(\boldsymbol{q})$. The primary changes are that (7.23) approximates the derivative with a finite difference; enforces Coulomb friction (7.23d) and inelasticity (7.23c) at the incremented velocity $\boldsymbol{v}'$; and replaces Coulomb friction with its linear approximation. However, computationally, our method seems most similar to the LCP-based algebraic method (2.29), and we will show the each increment (7.23) can also be solved as an LCP. Despite this similarity, the are significant philosophical differences between (2.29) and (7.23) that lead to qualitatively different predictions. As opposed to the unrealistic velocity-based complementarity constraint (2.29c), the termination condition $\boldsymbol{J}_{n,i}\boldsymbol{v}' \geq 0$ is removed from (7.23c), and thus it may take many increments of our model to reach post-impact velocity. Furthermore, the removal of this constraint makes (7.23) underconstrained, and thus allows significant freedom for selection of the normal impulse increments.

As our model intentionally captures a set realistic outcomes, we frame resolving an impact as sampling from that set. We parameterize the sampling process with a normal impulse distribution $p(\boldsymbol{\lambda}_n)$ over the unit box; step size $h > 0$; and (possibly infinite) max iteration count N. We compute samples from our discrete approximation of Routh's method (Algorithm 1) as follows:

1. Generate a non-zero, maximum normal impulse increment $\boldsymbol{\lambda}_{n,max} \sim h \cdot p(\boldsymbol{\lambda}_n)$.

2. Find a set of forces $\boldsymbol{\lambda} \neq \mathbf{0}$ with normal component $\boldsymbol{\lambda}_n \leq \boldsymbol{\lambda}_{n,max}$ that solves (7.23); Increment $\boldsymbol{v} \leftarrow \boldsymbol{v} + \boldsymbol{M}^{-1}\boldsymbol{J}^T\boldsymbol{\lambda}$.

3. Terminate and take $\boldsymbol{v}^+ = \boldsymbol{v}$ if it is non-colliding ($\boldsymbol{v} \notin \mathcal{C}(\boldsymbol{q})$) or the iteration limit is reached; else, return to 1.

While our theoretical results extend to any finite-density $p(\boldsymbol{\lambda}_n)$, we assume in this section that $p(\boldsymbol{\lambda}_n)$ is the uniform distribution for simplicity. For notational simplicity, we assume that all contacts are active and non-penetrating ($\boldsymbol{\phi}(\boldsymbol{q}) = 0$).

A difficulty in step 2) above is that $\boldsymbol{\lambda} = \boldsymbol{0}$ solves (7.23), and makes no progress towards impact termination. Additionally, it is possibile that no solution to step 2) allows $\boldsymbol{\lambda}_n = \boldsymbol{\lambda}_{n,max}$. We therefore add constraints that encourage $\boldsymbol{\lambda}_n$ to be large:

$$\boldsymbol{0} \leq \boldsymbol{\beta} \perp \boldsymbol{\lambda}_{n,max} - \boldsymbol{\lambda}_n \geq \boldsymbol{0}\,, \tag{7.24}$$

$$\boldsymbol{0} \leq \boldsymbol{\lambda}_n \perp \boldsymbol{J}_n \boldsymbol{v}' + \boldsymbol{\beta} \geq \boldsymbol{0}\,. \tag{7.25}$$

Together, (7.24)–(7.25) enforce (7.23c); $\boldsymbol{\lambda}_n \leq \boldsymbol{\lambda}_{n,max_i}$; and either $\boldsymbol{\lambda}_{n,i} = \boldsymbol{\lambda}_{n,max_i}$ or contact i has terminated.

Similar to the methods of Glocker and Pfeiffer (1995) and Anitescu and Potra (1997) described in Section 2.5.2 (see also (2.30)–(2.31)), we transcribe our model as LCP($\boldsymbol{W}_{\boldsymbol{q}_0}, \boldsymbol{w}_{\boldsymbol{q}_0}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max})$):

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{\beta} \\ \bar{\boldsymbol{\lambda}} \\ \boldsymbol{\gamma} \end{bmatrix}, \quad \boldsymbol{w}_{\boldsymbol{q}}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max}) = \begin{bmatrix} \boldsymbol{\lambda}_{n,max} \\ \bar{\boldsymbol{J}}\boldsymbol{v} \\ \boldsymbol{0} \end{bmatrix}, \tag{7.26a}$$

$$\boldsymbol{W}_{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{I} & \boldsymbol{J}_n(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}_n(\boldsymbol{q}) & \boldsymbol{J}_n(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}_D(\boldsymbol{q}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J}_D(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}_n(\boldsymbol{q}) & \boldsymbol{J}_D(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}_D(\boldsymbol{q}) & \boldsymbol{E} \\ \boldsymbol{0} & \boldsymbol{\mu} & -\boldsymbol{E}^T & \boldsymbol{0} \end{bmatrix}, \tag{7.26b}$$

$$\boldsymbol{v}'(\bar{\boldsymbol{\lambda}}) = \boldsymbol{v} + \boldsymbol{M}^{-1}\bar{\boldsymbol{J}}^T\bar{\boldsymbol{\lambda}}\,, \tag{7.26c}$$

where $\boldsymbol{q}$ is the configuration at impact; $\boldsymbol{\mu} = \mathrm{diag}(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m)$; and $\boldsymbol{E} = \mathrm{blkdiag}(\boldsymbol{1}, \ldots, \boldsymbol{1})$. We

note in particular that the columns and rows of $\boldsymbol{W_q}$ associated with $[\bar{\boldsymbol{\lambda}}; \boldsymbol{\gamma}]$ above are identical to the LCP matrix from Anitescu and Potra (1997).

Algorithm 1: $\mathrm{Sim}(h, \boldsymbol{x}_0, N)$

Input: step h, initial state $\boldsymbol{x}_0 = [\boldsymbol{q}_0; \boldsymbol{v}_0]$, max iterations N

Output: final velocity $\boldsymbol{v}$

$(\boldsymbol{v}, i) \leftarrow (\boldsymbol{v}_0, 0)$;

while $\boldsymbol{v} \in \mathcal{C}(\boldsymbol{q}_0)$ *and* $i \leq N$ **do**

  $\boldsymbol{\lambda}_{n,max} \sim h \cdot p(\boldsymbol{\lambda}_n)$;

  Select $\boldsymbol{z} = [\boldsymbol{\beta}; \bar{\boldsymbol{\lambda}}; \boldsymbol{\gamma}] \in \mathrm{LCP}(\boldsymbol{W}_{\boldsymbol{q}_0}, \boldsymbol{w}_{\boldsymbol{q}_0}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max}))$;

  $(\boldsymbol{v}, i) \leftarrow (\boldsymbol{v} + \boldsymbol{M}(\boldsymbol{q}_0)^{-1} \bar{\boldsymbol{J}}(\boldsymbol{q}_0)^T \bar{\boldsymbol{\lambda}}, i + 1)$;

end

7.4.2. Properties

Existence

The most essential property of our integration step is that, because $\boldsymbol{W_q}$ is copositive, we can leverage Proposition 3 to show that the constituent LCP has a solution:

Theorem 11 (Single-Step Existence (Appendix E.5.1))**.** $\mathrm{LCP}(\boldsymbol{W_q}, \boldsymbol{w_q}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max}))$ *is non-empty for all states* $[\boldsymbol{q};\ \boldsymbol{v}]$*, and normal impulse* $\boldsymbol{\lambda}_{n,max} \geq \boldsymbol{0}$*.*

Dissipation

As discussed in Section 7.2.3, an essential property of inelastic impacts is energy dissipation; because solutions to our model approximate the DI, each integration step cannot increase kinetic energy:

Theorem 12 (Dissipation (Appendix E.5.2))**.** *Let* $[\boldsymbol{q};\ \boldsymbol{v}]$ *be any state with active contact, and let* $\boldsymbol{\lambda}_{n,max} \geq \boldsymbol{0}$ *be a normal impulse. Then all impulses* $\bar{\boldsymbol{\lambda}}$ *generated by the impact constraints* $\mathrm{LCP}(\boldsymbol{W_q}, \boldsymbol{w_q}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max}))$ *dissipate kinetic energy:*

$$K(\boldsymbol{q}, \boldsymbol{v}'(\bar{\boldsymbol{\lambda}})) \leq K(\boldsymbol{q}, \boldsymbol{v})\,. \tag{7.27}$$

Impulse Advancement

If $\boldsymbol{\lambda}_n = \mathbf{0}$ were allowed by the LCP at a penetrating velocity $\boldsymbol{v} \in \mathcal{C}(\boldsymbol{q})$, then $\boldsymbol{v} = \boldsymbol{v}'$ could be selected in an infinite loop, and Algorithm 1 might never terminate. The structure of the normal impulse constraints (7.24) and (7.25) prevents this behavior by design for $\boldsymbol{\lambda}_{n,max} > \mathbf{0}$:

Lemma 7 (Impact Advancement (Appendix E.5.3))**.** *Let* $[\boldsymbol{q};\ \boldsymbol{v}]$ *be colliding (*$\boldsymbol{v} \in \mathcal{C}(\boldsymbol{q})$*), and* $\boldsymbol{\lambda}_{n,max} > \mathbf{0}$*. Let* $\bar{\boldsymbol{\lambda}} = [\boldsymbol{\lambda}_n;\ \boldsymbol{\lambda}_D]$ *be an impulse generated by* $\mathrm{LCP}(\boldsymbol{W}_{\boldsymbol{q}}, \boldsymbol{w}_{\boldsymbol{q}}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max}))$*. Then either some contact* i *activates fully (*$\boldsymbol{\lambda}_{n,i} = \boldsymbol{\lambda}_{n,max_i}$*), or all contacts terminate (*$\boldsymbol{J}_n \boldsymbol{v}'(\bar{\boldsymbol{\lambda}}) \geq \mathbf{0}$*).*

However, preventing the infinite loop also requires the stricter condition that the net impulse $\bar{\boldsymbol{J}}^T \bar{\boldsymbol{\lambda}}$ be non-zero. Assumption 2 in fact yields a stricter condition: the change in velocity $\boldsymbol{M}^{-1} \bar{\boldsymbol{J}}^T \bar{\boldsymbol{\lambda}}$ moves linearly in $\|\boldsymbol{\lambda}_n\|_1$ in a particular direction $\boldsymbol{r}(\boldsymbol{q})$:

Lemma 8 (Net Force Bound (Appendix E.5.4))**.** *Consider a configuration* $\boldsymbol{q} \in \mathcal{Q}_A \setminus \mathcal{Q}_P$*. There exists a nonzero* $\boldsymbol{r}(\boldsymbol{q}) \in \mathbb{R}^{n_v}$*, such that for any* $\bar{\boldsymbol{\lambda}} = [\boldsymbol{\lambda}_n;\ \boldsymbol{\lambda}_D]$ *obeying* (2.31)*,*

$$\boldsymbol{r}(\boldsymbol{q}) \cdot \boldsymbol{M}^{-1} \bar{\boldsymbol{J}}(\boldsymbol{q})^T \bar{\boldsymbol{\lambda}} \geq \|\boldsymbol{\lambda}_n\|_1 \ . \tag{7.28}$$

We briefly note that $\boldsymbol{r}(\boldsymbol{q})$ is computable as a linear program, as it arises from minimization over a polygonal set, LFC.

7.4.3. Linear Impact Termination

We now show that Algorithm 1 likely terminates in a small number of steps, allowing it to be used to implement $\mathrm{ImpactLaw}(\boldsymbol{q}, \boldsymbol{v})$. Let the random variable $Z(h, \boldsymbol{q}_0, \boldsymbol{v}_0)$ be the number of LCP solves required for $\mathrm{Sim}(h, [\boldsymbol{q}_0;\ \boldsymbol{v}_0], \infty)$ to terminate. Given that multiple impacts might occur in a single time-step, it is crucial that $Z(h, \boldsymbol{q}_0, \boldsymbol{v}_0)$ be as small as possible. Consider that Lemma 8 implies that the velocity takes large steps in the $\boldsymbol{r}$ direction with high probability, yet total movement in any direction is bounded by $2\|\boldsymbol{v}_0\|_{\boldsymbol{M}}$ as kinetic energy is non-increasing (Lemma 7). We can therefore show that with high probability, Z grows linearly with $\|\boldsymbol{v}_0\|_{\boldsymbol{M}}$:

Theorem 13 (Discrete Termination (Appendix E.5.5))**.** *Let $\boldsymbol{q}_0 \in \mathcal{Q}_A \setminus \mathcal{Q}_P$ have m active contacts. Pick σ such that $\boldsymbol{M}(\boldsymbol{q}_0) \succeq \sigma \boldsymbol{I}$; pick $\boldsymbol{r}(\boldsymbol{q}_0)$ as in Lemma 8; let $h > 0$ be a step-size. Let*

$$c = 4 \left\lceil \frac{(m+1)\|\boldsymbol{r}(\boldsymbol{q}_0)\|_2}{h\sqrt{\sigma}} \right\rceil . \tag{7.29}$$

Then for all $k \in \mathbb{Z}^+$, $\boldsymbol{v}_0 \in \mathcal{C}(\boldsymbol{q}_0)$,

$$P\left(Z(h, \boldsymbol{q}_0, \boldsymbol{v}_0) > c \left\lceil \|\boldsymbol{v}_0\|_{\boldsymbol{M}} \right\rceil + k\right) \leq e^{-\frac{k}{(m+1)^2}} . \tag{7.30}$$

As the probability density of Z exponentially decays, it has finite moments (including mean and variance).

7.4.4. Post-Impact Set Approximation

We now describe a method to approximate the set of outcomes to simultaneous impacts as modeled in our DI (7.3), which culminates in probabilistic guarantees on densely sampling this set via Proposition 5.

In order for computation of the set of possible outcomes of Algorithm 1 to be well-posed, we must consider a key practical ramification of the LCP solve on line 4: numerical LCP solvers typically only find a *single* solution, and may be systematically biased in their selection among multiple solutions. For all claims in this section, we therefore make the additional assumption that this selection process does not affect the outcome of an individual integration step:

Assumption 7. *Consider a configuration $\boldsymbol{q} \in \mathcal{Q}_A \setminus \mathcal{Q}_P$. For each velocity $\boldsymbol{v}$ and normal impulse increment $\boldsymbol{\lambda}_{n,max} \geq \boldsymbol{0}$, every $\bar{\boldsymbol{\lambda}}$ generated by* (7.26) *results in the same incremented velocity $\boldsymbol{v}'$. Equivalently, there exists a function $\boldsymbol{f}_{\boldsymbol{q}} : \mathbb{R}^{n_v} \times \mathrm{cl}(\mathbb{R}^{m+}) \to \mathbb{R}^{n_v}$, such that*

$$\boldsymbol{v}'(\bar{\boldsymbol{\lambda}}) = \boldsymbol{v} + \boldsymbol{M}^{-1}\bar{\boldsymbol{J}}^T\bar{\boldsymbol{\lambda}} = \boldsymbol{f}_{\boldsymbol{q}}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max}) . \tag{7.31}$$

This assumption can be verified via Semidefinite Programming (Aydinoglu et al., 2022). We note that $\boldsymbol{v}'(\bar{\boldsymbol{\lambda}})$ under Assumption 7 is only unique *given* $\boldsymbol{\lambda}_{n,max}$; different velocity increments can be While Assumption 7 is violated for at least some systems (e.g. for compass gait and RAMone in Section 7.5), it implies useful properties including Lipschitz continuity:

Lemma 9. *For each configuration $\boldsymbol{q} \in \mathcal{Q}_A \setminus \mathcal{Q}_P$, $\boldsymbol{f}_{\boldsymbol{q}}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max})$ is Lipschitz continuous.*

Proof. Because $\boldsymbol{v}'(\bar{\boldsymbol{\lambda}})$ is unique, we must have that

$$\bar{\boldsymbol{J}}^T \bar{\boldsymbol{\lambda}} = \begin{bmatrix} \boldsymbol{0} & \bar{\boldsymbol{J}}^T & \boldsymbol{0} \end{bmatrix} \mathrm{LCP}(\boldsymbol{W}_{\boldsymbol{q}}, \boldsymbol{w}_{\boldsymbol{q}}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max}))\,, \tag{7.32}$$

is a singleton over the convex domain $\boldsymbol{w}_{\boldsymbol{q}}(\mathbb{R}^{n_v}, \mathrm{cl}(\mathbb{R}^{m+}))$. Therefore by Proposition 4, $\boldsymbol{f}_{\boldsymbol{q}}$ is Lipschitz continuous. □

We will also make use of two scenarios where the integration step LCP is guaranteed select zero impulse:

Lemma 10. *Consider a configuration $\boldsymbol{q} \in \mathcal{Q}_A \setminus \mathcal{Q}_P$ and $\boldsymbol{\lambda}_{n,max} \geq \boldsymbol{0}$. Then if either $\boldsymbol{J}_n \boldsymbol{v} \geq \boldsymbol{0}$ or $\boldsymbol{\lambda}_{n,max} = \boldsymbol{0}$,*

$$\boldsymbol{v} = \boldsymbol{f}_{\boldsymbol{q}}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max})\,. \tag{7.33}$$

Proof. Observe that if either $\boldsymbol{\lambda}_{n,max} = \boldsymbol{0}$ or if $\boldsymbol{v}$ is not impacting ($\boldsymbol{J}_n \boldsymbol{v} \geq \boldsymbol{0}$), we can select zero normal impulse ($\boldsymbol{\lambda}_n = \boldsymbol{0}$, thus $\boldsymbol{v}' = \boldsymbol{v}$) and satisfy the normal complementary equations (7.24)–(7.25). Setting $\boldsymbol{\lambda}_D = \boldsymbol{0}$; $\boldsymbol{\gamma} = \boldsymbol{0}$; and $\boldsymbol{\beta}$ as the negative part of $\boldsymbol{J}_n \boldsymbol{v}$ constitutes a full solution to the LCP. □

The continuity of $\boldsymbol{f}_{\boldsymbol{q}}$ allows for expansion of the $\boldsymbol{J}_n \boldsymbol{v} \geq \boldsymbol{0}$ case; if $\boldsymbol{v}$ is *almost* terminated, then only a single simulation step with a small $\boldsymbol{\lambda}_{n,max}$ is required to end the impact:

Lemma 11 (Single-Step Termination (Appendix E.5.6))**.** *For all configurations $\boldsymbol{q} \in \mathcal{Q}_A \setminus \mathcal{Q}_P$, velocities $\boldsymbol{v}$, and $\varepsilon > 0$, there exists $\delta(\varepsilon, \boldsymbol{v})$, such that for any almost-separating velocity $\bar{\boldsymbol{v}}$ ($\boldsymbol{J}_n\bar{\boldsymbol{v}} \geq -\delta(\varepsilon, \boldsymbol{v})$) that is sufficiently small ($\|\bar{\boldsymbol{v}}\|_{\boldsymbol{M}} \leq \|\boldsymbol{v}\|_{\boldsymbol{M}}$), a small impulse can terminate the impact: $\boldsymbol{f}_{\boldsymbol{q}}(\bar{\boldsymbol{v}}, \varepsilon\mathbf{1}) \notin \mathcal{C}(\boldsymbol{q})$.*

We now iteratively define the reachable set of Alg. 1. Let $\mathcal{V}_N(\boldsymbol{x}_0, h)$ be the set of possible outputs of $\mathrm{Sim}(h, \boldsymbol{x}_0, N)$. Then we have that

$$\mathcal{V}_0(\boldsymbol{x}_0, h) = \{\boldsymbol{v}_0\}\ , \tag{7.34}$$

$$\mathcal{V}_N(\boldsymbol{x}_0, h) = \boldsymbol{f}_{\boldsymbol{q}}(\mathcal{V}_{N-1}(\boldsymbol{x}_0, h), [0, h]^m)\,, \tag{7.35}$$

$$\mathcal{V}_N(\boldsymbol{x}_0, h) \supseteq \mathcal{V}_{N-1}(\boldsymbol{x}_0, h)\,. \tag{7.36}$$

Here, we used Lemma 10 to ignore early termination (i.e. $\boldsymbol{J}_n\boldsymbol{v} \geq \mathbf{0}$ before N loop iterations) in (7.35), and to establish the monotonic growth in (7.36). We construct the entire set of reachable velocities as

$$\mathcal{V}_\infty(\boldsymbol{x}_0, h) = \cup_{N\in\mathbb{N}}\mathcal{V}_N(\boldsymbol{x}_0, h)\,. \tag{7.37}$$

$\mathcal{V}_N(\boldsymbol{x}_0, h)$ can approximate $\mathcal{V}_\infty(\boldsymbol{x}_0, h)$ with arbitrarily well:

Lemma 12. *Consider a configuration $\boldsymbol{q}_0 \in \mathcal{Q}_A \setminus \mathcal{Q}_P$; velocity $\boldsymbol{v}_0$, and step-size $h \geq 0$. Then for each $\varepsilon > 0$, there exists an N, such that $\mathcal{V}_N([\boldsymbol{q}_0;\ \boldsymbol{v}_0], h)$ is an ε-net of $\mathcal{V}_\infty([\boldsymbol{q}_0;\ \boldsymbol{v}_0], h)$.*

Proof. $\mathcal{V}_N([\boldsymbol{q}_0;\ \boldsymbol{v}_0], h)$ is a monotonic (7.36) and uniformly bounded (via Theorem 12) sequence of sets. It is then convergent in the ε-net sense to $\cup_N\mathcal{V}_N = \mathcal{V}_\infty([\boldsymbol{q}_0;\ \boldsymbol{v}_0], h)$. □

Similarly, the post-impact reachable set is simply the reachable velocities which are non-penetrating:

$$\mathrm{Sim}(h, \boldsymbol{x}_0, \infty) \in \mathcal{V}_\infty(\boldsymbol{x}_0, h) \setminus \mathcal{C}(\boldsymbol{q}_0)\,. \tag{7.38}$$

Algorithm 2: Approximate$(h, \boldsymbol{x}_0, \varepsilon, N, M)$

Input: step size h, initial state $\boldsymbol{x}_0 = [\boldsymbol{q}_0;\ \boldsymbol{v}_0]$, approximation $\varepsilon \in (0, h)$, trajectory length N, trajectory count M

Output: post-impact set approximation $\tilde{\mathcal{V}}^+$

$\tilde{\mathcal{V}}^+ \leftarrow \{\}$;

$\psi \leftarrow \sigma_{max}\left(\boldsymbol{M}^{-1}\bar{\boldsymbol{J}}^T\right) m(1 + \max_i \boldsymbol{\mu}_i) + 1$;

for $i = 1$ **to** M **do**

  $\boldsymbol{v} \leftarrow \text{Sim}(h, \boldsymbol{x}_0, N)$;

  $\tilde{\mathcal{V}}^+ \leftarrow \tilde{\mathcal{V}}^+ \cup \left\{\boldsymbol{f}_{\boldsymbol{q}_0}(\boldsymbol{v}, \frac{\varepsilon}{3\psi}\mathbf{1}_m)\right\}$;

end

$\tilde{\mathcal{V}}^+ \leftarrow \tilde{\mathcal{V}}^+ \setminus \mathcal{C}(\boldsymbol{q}_0)$;

We can finally use the above derived properties to construct a method, Algorithm 2, for approximating the post-impact set. Lemma 12 and Proposition 5 together show that M samples from $\text{Sim}(h, \boldsymbol{x}_0, N)$ well-approximate $\mathcal{V}_\infty$, and can be forced to terminate with only a small additional step (Lemma 11). Therefore, Algorithm 2 is approximately complete:

Theorem 14. *Consider an initial configuration $\boldsymbol{q}_0 \in \mathcal{Q}_A \setminus \mathcal{Q}_P$, initial velocity $\boldsymbol{v}_0 \in \mathbb{R}^{n_v}$, and step-size $h > 0$. For all $\varepsilon, \delta > 0$, there exists $N, M > 0$, such that* Approximate$(h, \boldsymbol{x}_0, \varepsilon, N, M)$ *returns an ε-net of $\mathcal{V}_\infty(\boldsymbol{x}_0, h) \setminus \mathcal{C}(\boldsymbol{q}_0)$ with probability at least $1 - \delta$.*

Proof. See Appendix E.5.7 □

7.5. Numerical Examples

We now show several examples of the post-impact velocity sets generated by our model; complete details are provided in Appendix section E.1. The MATLAB code is available online[7], and LCP's were solved via the PATH solver (Dirkse and Ferris, 1995). We analyze the three examples shown thus far (Figs. 7.1, 7.2, 7.3), along with two more complex systems. For each system, we plot the evolution of the velocity through the impact process with lines, projected onto the contact

[7] https://github.com/mshalm/routh-multi-impact

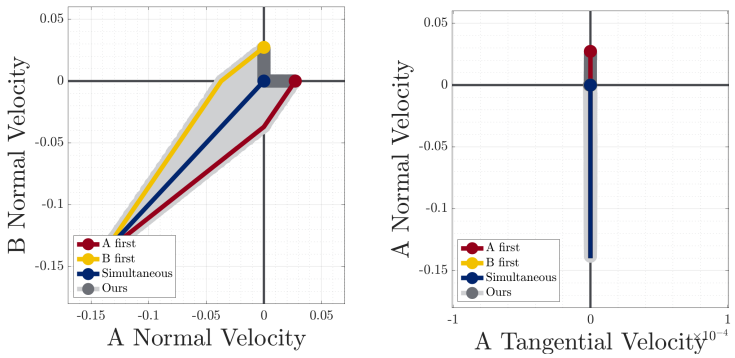


Figure 7.5: Evolution of the block-ground impact, displayed as the normal velocities of the two points (top), and the normal and tangential velocity of point A (bottom). Our method generates all three outcomes from Figure 7.1, as well as intermediate results between the symmetric and sequential impacts.

frames. Our method is shown in gray and simultaneous resolution via Anitescu and Potra (1997) is shown in blue. For two-contact impacts, we show the two sequential resolutions $(A, B, A, \dots)$ and $(B, A, B, \dots)$in red and yellow. We show samples of the post-impact velocity sets generated via Algorithm 2, as dark gray circles. For some examples, axes of symmetry were used to duplicate samples.

7.5.1. Block Drop

We revisit dropping a narrow, rectangular object onto flat ground (Fig. 7.1), which may either result in the object coming to rest or pivoting on a corner. As shown in Figure 7.5, our method produces each of these symmetric and sequential outcomes. The real-world analogues of these three outcomes are that the short but non-zero duration impacts either happen at the exact same time and rate or sequentially with no overlap. Our model also produces analogues to where there is some partial overlap in these durations, for which scaled-down versions of the purely-sequential outcomes (i.e., rolling on one foot with a smaller angular velocity) is the final result.

7.5.2. Compass Gait

We revisit the compass gait walker model taking a wide step (Fig. 7.2). Previously, we showed that the model of Anitescu and Potra (1997) predicts that the leading foot sticks (point A), while the trailing foot (point B) could slide, stick, or lift off. Our model generates each of these outcomes,

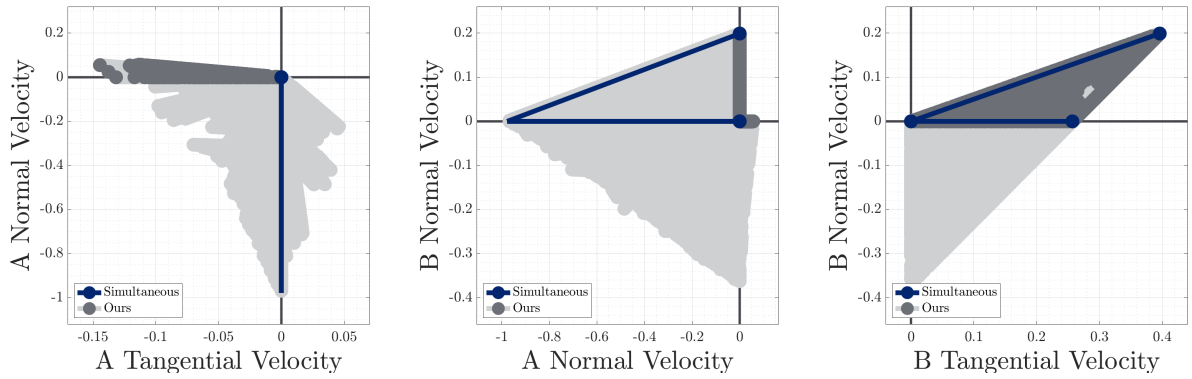


Figure 7.6: Evolution of the compass gait step. The center plot compares the normal velocities of the two contacts, while the left and right show velocities of points A and B, respectively. Our model produces the three outcomes in Figure 7.2b, as well as all reasonable intermediate velocities of point B. Furthermore, oscillation of impact between the feet allows point A to slide or lift off, while point B maintains contact.

as well as various convex combinations of these results (Figure 7.6). It also generates oscillatory behavior where impulses at points A and B alternate during the impact process. This can potentially cause A to lift off, and B to remain on the ground instead.

7.5.3. Box and Wall

We examine our model's predictions on the scenario described in Figure 7.3, where a box impacts a wall (at point B) while sliding along flat ground (at point A). Simultaneous resolution with Anitescu and Potra (1997) predicted that the box came to rest, while sequential resolution predicts that A continues sliding and B lifts off. As in the previous examples, our model reproduces both behaviors, as well as convex combinations of them (Figure 7.7). Additionally, some sequences allow A to slide even faster, while others allow B to slide instead of lifting or sticking.

7.5.4. RAMone

In this example, we examine a footstep of a more complex 5-link bipedal robot, RAMone, originally considered by Remy (2017). As shown in Figure 7.8, much like the compass gait example, Anitescu and Potra (1997) always predicts that the leading foot sticks, while the trailing foot can stick, slide, or lift. Our model reproduces the same results, as well as ones where the final contact velocities are scaled down.

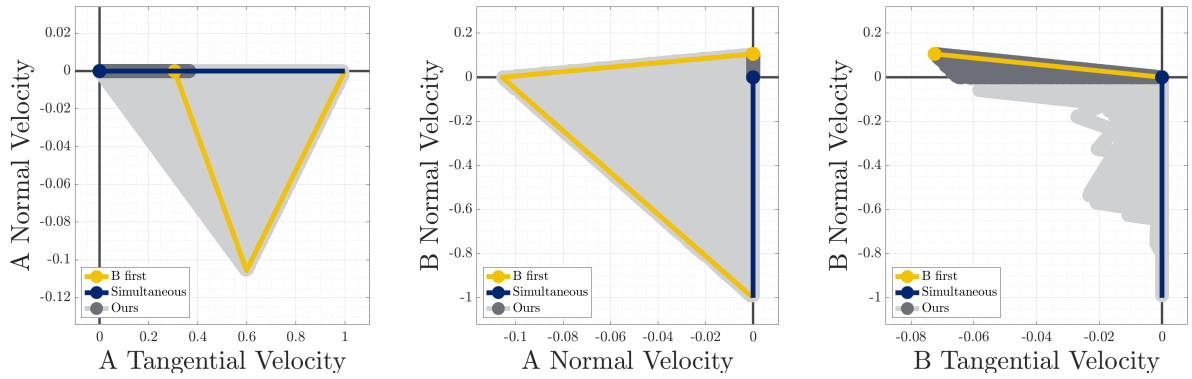


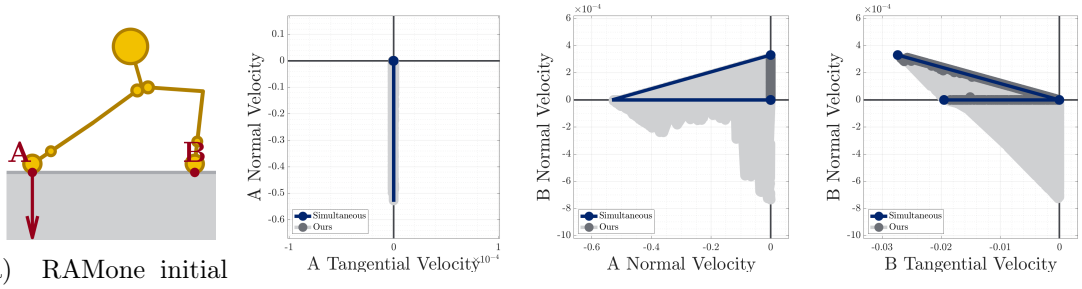
Figure 7.7: Evolution of the box-wall impact. The center plot compares the normal velocities of the two contacts, while the left and right show velocities of points A and B, respectively. Our model produces both the simultaneous and sequential outcome in Figure 7.3, as well as all reasonable intermediate velocities where A still slides and B lifts off. Furthermore, results are also generated where B slides instead of sticking or lifting off.

7.5.5. Disk Stacking

In this example, we demonstrate our ability to generate non-unique results in a multi-object scenario motivated by manipulation: stacking disks. A tower of 3 discs (Figure 7.9) is created by dropping a disk on two others, which rest on the ground. The only prediction for this 5-contact collision offered by Anitescu and Potra (1997) is the entire tower coming to rest. While we cannot be sure that the numerical results cover all possible outputs of our model, we are able to generate various outcomes in which the tower falls apart. Figure 7.9b shows how the post-impact normal velocities compare in the left and right sides of the tower. The top ball always maintains contact with at least one of the left or right balls, and one of those balls always stays on the ground.

7.6. Discussion

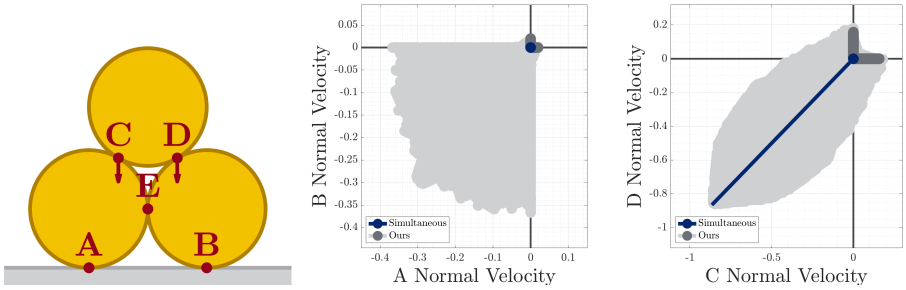
Non-unique behavior is a pervasive complexity that is present in both real-world robotic systems and common models capturing frictional impacts between rigid bodies—and thus accurate incorporation of such phenomena is an essential component of robust planning, control, and estimation algorithms. Our model presents a state-of-the-art theoretical foundation for capturing these set-valued outcomes. Despite the high versatility of allowing impacts to resolve at arbitrary relative rates, both the continuous-time formulation and simulation method have termination guarantees.



(a) RAMone initial condition

(b) Footfall impact resolution process

Figure 7.8: Evolution of a footfall (a) of the RAMone robot. (b) Similar to the compass gait example, Anitescu and Potra (1997) predicts that the leading foot, point A, comes to rest, while point B may come to rest, slide, or lift off. All results from our model produce intermediate outcomes between these three results, and point A remains in stiction for the entire duration of the impact.



(a) Disk stack initial condition

(b) Normal impact resolution process comparing left and right sides

Figure 7.9: Evolution of a stack of disks (a) as the top disk fall on the bottom two. Anitescu and Potra (1997) only predicts that the entire system comes to rest. (b) Our method additional predicts several scenarios where the top disk remains in contact with only one of the bottom two disks, while the other may roll away or even lift off the ground slightly.

Future development of our model will focus on capturing a wider array of contact-driven behaviors; improved theoretical guarantees; and more efficient computational approaches. For instance, while many models in robotics assume impacts are inelastic, capturing restitution would increase the accuracy of our model for some systems. Additionally, while approximation of the post-impact set is probabilistically complete, there is currently no available method to compute the provided bounds. Furthermore, the he bounds were not tight enough to inform tractable computation in our examples. Future research could develop outer approximations of the post-impact set via Lyapunov reachability and semidefinite programming (Posa et al., 2016b).

CHAPTER 8

Conclusions and Future Work

In this thesis, we examined both diagnoses of and solutions to challenges in accurately modeling the dynamics of rigid robots interacting with their environments through rigid, physical contact. We find that perspectives from mechanics, optimization, and machine learning can be combined to illuminate numerical stiffness as a core challenge to this end. By combining mathematical tools from each of these areas, we have presented advancements in both parameter identification and theoretical modeling of the behaviors of these systems. Our contributions and complementary recent work provide strong evidence that physics-based modeling is a strong paradigm from robotic identification, simulation, planning, and control (Jin and Posa, 2022; Aydinoglu et al., 2023).

While the entirety of this work has focused on rigid-body systems, and many rigid robots are currently in use, there a growing focus both on manipulation of soft, deformable objects as well as robots that are themselves deformable. Such systems are still governed by physical laws which may be captured as numerical optimization (Han et al., 2023), where numerical stiffness is not only present, but also directly captured as mechanical stiffness. In addition to high-dimensional unknown parameters, the state spaces of deformable objects are also in general infinite-dimensional, posing additional computational challenges. Important questions for future work will center on the necessity, tractability, and construction of methods similar to our proposed work.

Additionally, high-fidelity sensing is becoming more accessible in the form of RGB(D) sensors, sometimes integrated with contact sensing in *visuotactile* sensors (Taylor et al., 2022). These devices provide rich, local observation not only of object geometries, but also of the contact forces between manipulands and robots. It is an open question how such data my be best utilized for warm-starting or sensor fusion with our geometric representations, potentially enabling faster training convergence and higher-quality geometry estimation. Additionally, while rigid body models may struggle to predict a single, accurate set of contact impulses in response to a collision, hardware which can indirectly observe the impulses themselves may prove a particularly useful tool for data-driven

modeling.

APPENDIX A

ADDITIONAL THEORETICAL BACKGROUND

In this appendix, we discuss detailed theoretical background and technical lemmas necessary for detailed understanding of the theoretical results in this thesis. We begin with some essential results from functional analysis (Section A.1). We then discuss generalizations of derivative calculus (Section A.2) and properties of associated inequalities (Section A.3) used in our results in Chapter 5. We conclude with some technical lemmas on bilevel optimization in support of the proofs in Appendix C (Section A.4).

A.1. Functional Analysis

The results in this work are derived using tools from measure theory and functional analysis; for a thorough background, see Rudin (1986, 1991). We equip domains $\Omega \subseteq \mathbb{R}^n$ with the Euclidean metric and norm, and the Lebesgue measure by default. The total derivative $\dot{\boldsymbol{f}}(t)$ of an absolutely continuous function $\boldsymbol{f}(t)$ is taken in the Lebesgue sense. We make use of the following theorem:

Theorem 15 (Arzelà-Ascoli (Rudin, 1991))**.** *Every uniformly bounded sequence* $\{\boldsymbol{f}_n\}_{n\in\mathbb{N}}$ *of equicontinuous* $\mathbb{R}^n$*-valued functions on a compact interval* $[a, b]$ *has a subsequence* $(\boldsymbol{f}_{n_k})_{k\in\mathbb{N}}$ *that converges uniformly.*

$\{\boldsymbol{f}_n\}_{n\in\mathbb{N}}$ is equicontinuous if, for instance, each $\boldsymbol{f}_n$ is Lipschitz with the same constant.

A.2. Generalizations of Derivative Calculus

Our theoretical results for Chapter 5 rely on various generalizations of Lebesgue gradients (which we denote with the operator ∇), each equipped with their own calculus, e.g., addition and chain rules. We describe the foundations of each used in this thesis.

A.2.1. Fréchet Subgradients

Freéchet subgradients are a type of gradient generalization for non-smooth functions.

Definition 7. *The Fréchet subgradient* $\partial f(\boldsymbol{z}) : \mathbb{R} \to \mathbb{P}(\mathbb{R}^n)$ *of a function* $f(\boldsymbol{x}) : \mathbb{R} \to \mathbb{R}^n$*, is a set*

valued make containing exactly the solutions $\boldsymbol{s}$ *to*

$$\liminf_{\boldsymbol{y}\to\boldsymbol{z},\boldsymbol{y}\neq\boldsymbol{z}} \frac{f(\boldsymbol{y}) - f(\boldsymbol{z}) - (\boldsymbol{y}-\boldsymbol{z})\cdot\boldsymbol{s}}{\|\boldsymbol{y}-\boldsymbol{z}\|_2} \geq 0\,. \tag{A.1}$$

f is said to be subdifferentiable at $\boldsymbol{x}$ when $\partial f(\boldsymbol{x})$ is non-empty.

Basic Fréchet Subgradient Calculus

Fréchet subgradients have a similar calculus to gradients, especially for semiconvex functions:

Definition 8 (Bolte et al. (2010))**.** *A semiconvex function $f(\boldsymbol{z}) : \mathbb{R} \to \mathbb{R}^n$ is one for which there exists $\beta > 0$ such that $f(\boldsymbol{z}) + \beta\|\boldsymbol{z}\|_2^2$ is convex. Semiconcave functions are equal to the negative of a semiconvex function (Albano, 1999).*

We note the following rules for the Fréchet subgradients of various functions $f(\boldsymbol{z}), f_1(\boldsymbol{z}), f_2(\boldsymbol{z})$:

- ∂f is always closed and convex (Kruger, 2003, Section 1.1).
- Semiconvex functions are subdifferentiable everywhere. Their subgradients take compact values, and are upper semi-continuous in $\boldsymbol{z}$ (Albano, 1999, Section 3).
- When f is continuously differentiable at x, $\partial f = \nabla f$ (Karimi et al., 2016, Appendix G).
- if f is Lipschitz on a bounded open domain with constant L, then $\|\partial f\|_2 \leq L$ on that domain (Kruger, 2003, Proposition 1.17).
- When $f = f_1 + f_2$ on a neighborhood of x, $\partial f = \partial f_1 + \partial f_2$ when both f_1 and f_2 are subdifferentiable at x and:
  - one of the two is continuously differentiable (Kruger, 2003, Corollary 1.12.2); or
  - both are convex (Bertsekas, 2016, Proposition B.24).

We prove for ourselves a version of the chain rule for subgradients:

Theorem 16. *let $f(\boldsymbol{z}) = g(\boldsymbol{h}(\boldsymbol{z}))$ with g semiconvex in $\boldsymbol{h}$, and $\boldsymbol{h}$ continuously differentiable in $\boldsymbol{z}$.*

Then

$$\partial f(\boldsymbol{z}) = \partial g(\boldsymbol{h}(\boldsymbol{z}))\nabla \boldsymbol{h}(\boldsymbol{z})\,. \tag{A.2}$$

Proof. Both $\partial f(\boldsymbol{z})$ and $\partial g(\boldsymbol{h}(\boldsymbol{z}))\nabla \boldsymbol{h}(x)$ are closed and convex. Furthermore as g is semi-convex and continuous, ∂g is also compact. Furthermore, from Kruger (2003, Corollary 1.13.1), as $\boldsymbol{h}$ is differentiable, it is also locally Lipschitz, and

$$\partial f(\boldsymbol{z}) \supseteq \partial g(\boldsymbol{h}(\boldsymbol{z}))\partial \boldsymbol{h}(\boldsymbol{z})\,.$$

The reverse inclusion will therefore suffice. As both sets are convex and the latter is compact, it is sufficient to use a supporting hyperplane-type argument. That is, for all $\boldsymbol{f}^* \in \partial f(x), \|\boldsymbol{y}\| = 1\,,$ there exists $\boldsymbol{g}^* \in \partial g(\boldsymbol{h}(\boldsymbol{z}))$ such that

$$\boldsymbol{g}^* \nabla \boldsymbol{h}(\boldsymbol{z}) \cdot \boldsymbol{y} \geq \boldsymbol{f}^* \cdot \boldsymbol{y}\,.$$

Pick $\boldsymbol{f}^*, \boldsymbol{y}$ as such. Then by the definition of Fréchet subdifferential, for any sequence $\alpha_k \to 0$, $\alpha_k \neq 0$,

$$\liminf_{k\to\infty} \frac{g(\boldsymbol{h}(\boldsymbol{z}+\alpha_k \boldsymbol{y})) - g(\boldsymbol{h}(\boldsymbol{z})) - (\alpha_k \boldsymbol{y}) \cdot \boldsymbol{f}^*}{\alpha_k} \geq 0\,.$$

Pick $\beta > 0$ such that $g(\boldsymbol{h}) + \frac{\beta}{2}\|\boldsymbol{h}\|_2^2$ is convex in $\boldsymbol{h}$. By the semi-convexity of g and the definition of a convex subdifferential, we can select a bounded sequence $\boldsymbol{g}_k^* \in \nabla g(\boldsymbol{h}(\boldsymbol{z}+\alpha_k \boldsymbol{y}))$ with

$$g(h(\boldsymbol{z}+\alpha_k \boldsymbol{y})) - g(\boldsymbol{h}(\boldsymbol{z})) \leq \boldsymbol{g}_k^* \cdot (\boldsymbol{h}(\boldsymbol{z}+\alpha_k \boldsymbol{y}) - \boldsymbol{h}(\boldsymbol{z})) + \beta \left\|\boldsymbol{h}(\boldsymbol{z}+\alpha_k \boldsymbol{y}) - \boldsymbol{h}(\boldsymbol{z})\right\|^2\,.$$

As $\boldsymbol{h}$ is locally Lipschitz and differentiable,

$$\lim \frac{\beta \left\|\boldsymbol{h}(\boldsymbol{z}+\alpha_k \boldsymbol{y}) - \boldsymbol{h}(\boldsymbol{z})\right\|^2}{\alpha_k} = 0\,.$$

Combining the last three equations,

$$\liminf_{k\to\infty} \boldsymbol{g}_k^* \frac{\boldsymbol{h}(\boldsymbol{z}+\alpha_k \boldsymbol{y}) - \boldsymbol{h}(\boldsymbol{z})}{\alpha_k} - \boldsymbol{y}\cdot\boldsymbol{f}^* \geq 0\,.$$

WLOG by dropping to a subsequence $\boldsymbol{g}_k^*$ converges to $\boldsymbol{g}^* \in \partial g(\boldsymbol{h}(\boldsymbol{z}))$ by the upper semi-continuity of ∂g. Thus the sequence is composed of the product of two limiting sequences (as $\boldsymbol{h}$ is differentiable). Thus taking the limit we recover

$$\boldsymbol{g}^* \nabla \boldsymbol{h}(\boldsymbol{z})\cdot\boldsymbol{y} - \boldsymbol{f}^*\boldsymbol{y} \geq 0\,.$$

□

Examples

Consider the positive and negative parts of a vector v, i.e.

$$v = \mathrm{pos}(v) - \mathrm{neg}(v)\,. \tag{A.3}$$

Denote the subgradients $P(x) = \partial_x \mathrm{pos}(x)$ and $N(x) = \partial_x \mathrm{neg}(x)$. We have that

$$P(x) = \begin{cases} 0 & x < 0\,, \\ 1 & x > 1\,, \\ [0,1] & x = 0\,, \end{cases} \tag{A.4}$$

and $N(x) = \partial_x(\mathrm{pos}(x) - x) = P(x) - 1$. We also have the identities

$$\mathrm{pos}(x)P(x) = \mathrm{pos}(x)\,, \tag{A.5}$$

$$\mathrm{neg}(x)N(x) = \mathrm{neg}(x)\,, \tag{A.6}$$

$$\mathrm{pos}(x)N(x) = 0\,, \tag{A.7}$$

$$\mathrm{neg}(x)P(x) = 0\,. \tag{A.8}$$

Finally, we note that

$$\partial_x \frac{1}{2}\mathrm{neg}(x)^2 = -\mathrm{neg}(x)\,. \tag{A.9}$$

Consider now the convex function $v \to \|v\|_2$. We have that

$$\partial_v \|v\|_2 = \mathrm{Unit}(v) = \begin{cases} \hat{v} & v \neq 0\,, \\ \mathrm{clBall}(1) & v = 0\,. \end{cases} \tag{A.10}$$

A.2.2. Strong and Limiting Slope

We define the strong slope of any continuous function as a bound on the local rate of steepest descent:

Definition 9. *The strong slope $\|\nabla\| f$ of any continuous function f is well-defined and equal to*

$$\|\nabla\| f(\boldsymbol{z}) = \limsup_{\boldsymbol{y}\to\boldsymbol{z}, \boldsymbol{y}\neq\boldsymbol{z}} \frac{\mathrm{neg}(f(\boldsymbol{y}) - f(\boldsymbol{z}))}{\|\boldsymbol{y} - \boldsymbol{z}\|_2}\,. \tag{A.11}$$

When f is semiconvex, $\min_{\boldsymbol{s}\in\partial f} \|\boldsymbol{s}\| = \|\nabla\| f$ (Bolte et al., 2010, Lemma 43).

Definition 10. *The limiting slope $\|\nabla\| f$ of any continuous function f is well-defined and equal to*

$$\overline{\|\nabla\|} f(\boldsymbol{z}) = \liminf_{\boldsymbol{y}\to\boldsymbol{z}} \|\nabla\| f(\boldsymbol{y})\,. \tag{A.12}$$

Intuitively, the strong slope defines a pointwise rate of descent, and the limiting slope provides a local, pessimistic bound on that descent rate. It is well known that there exist *curves* of limiting descent under some regularity assumptions:

Theorem 17 (Near-steepest descent curves (Drusvyatskiy et al., 2015))**.** *Let $f(x)$ be continuous. Assume there exists η, r, K such that*

- $\eta < Kr$

- $\{\boldsymbol{z} : f(\boldsymbol{z}) < f(\boldsymbol{z}_0) - \eta\}$ *is non-empty.*
- $\{x : f(\boldsymbol{z}_0) - \eta < f(\boldsymbol{z}) \leq f(\boldsymbol{z}_0)\} \cap (\boldsymbol{z}_0 + \mathrm{clBall}_K) \subseteq \{\boldsymbol{z} : \|\nabla\| f(\boldsymbol{z}) \geq r\}$

  (i.e., the strong slope is large when $\boldsymbol{z}$ *is near* $\boldsymbol{z}_0$ *and* $f(\boldsymbol{z})$ *is closely below* $f(\boldsymbol{z}_0)$*)*

Then, there exists a near-steepest descent curve, i.e an $L > 0$ *and a 1-Lipschitz* $\boldsymbol{z}(t) : [0, L] \to \mathbb{R}^n$ *such that:*

- $\boldsymbol{z}(0) = \boldsymbol{z}_0$
- $\frac{\mathrm{d}}{\mathrm{d}t} f(\boldsymbol{z}(t)) \leq -\overline{\|\nabla\|} f(\boldsymbol{z}(t))$ *a.e.*
- $f(\boldsymbol{z}(L)) \leq f(\boldsymbol{z}_0) - \eta$.

A.3. Quadratic Growth and Descent Dominance

In the theoretical results for Chapter 5, we will use relationships between the growth and descent rates of various functions $f(\boldsymbol{z})$ with minimal value $\min_{\boldsymbol{z}} f(\boldsymbol{z}) = 0$. In particular, we focus on the definition of k-Quadratic Growth:

Definition 11. *A function* $f(\boldsymbol{z})$ *is* k*-Quadratic Growth (*k*-QG) if the following global inequality holds:*

$$f(\boldsymbol{z}) - \min f \geq \frac{k}{2} \operatorname{Dist}(\boldsymbol{z}, \arg\min f)^2 \,. \tag{A.13}$$

Intuitively a k-QG function is large for inputs far from its minimizers. Conversely, if the function is small for a given input, then the input is close to a minimizer:

$$\operatorname{Dist}(\boldsymbol{z}, \arg\min f) \leq \sqrt{\frac{k}{2} (f(\boldsymbol{z}) - \min f)} \,. \tag{A.14}$$

One might alternatively intuit that low function values indicate that a minimizer is close if the function can be descended quickly. We refer to this property as k-Descent Dominance:

Definition 12. *A function* $f(\boldsymbol{z})$ *is* k*-Descent Dominant (*k*-DD) if the following global inequality*

holds:

$$\overline{\|\nabla\|} f(\boldsymbol{z}) \geq \sqrt{2kf(\boldsymbol{z})}\,. \tag{A.15}$$

In fact, k-DD is a stronger condition condition than k-QG:

Theorem 18. *Let $f(\boldsymbol{z})$ with $\min_{\boldsymbol{z}} f(\boldsymbol{z}) = 0$. Then f k-DD implies f is k-QG.*

Proof. Consider $\boldsymbol{z}_0 \in \mathbb{R}^n$. We first estimate the distance of $\boldsymbol{z}_0$ to $\left\{\boldsymbol{z} : f(\boldsymbol{z}) \leq \frac{f(\boldsymbol{z}_0)}{2^{2n+1}}\right\}$. Consider $\eta = \left(1 - 2^{-2n-1}\right) f(\boldsymbol{z}_0)$. Then $\{\boldsymbol{z} : f(\boldsymbol{z}) < f(\boldsymbol{z}_0) - \eta\}$ is non-empty as $\min_{\boldsymbol{z}} f(\boldsymbol{z}) = 0$. Next, consider that $f(\boldsymbol{z}) \geq \frac{f(\boldsymbol{z}_0)}{2^{2n+1}}$ implies

$$\|\nabla\| f(\boldsymbol{z}) \geq \overline{\|\nabla\|} f(\boldsymbol{z}) \geq \sqrt{2kf(\boldsymbol{z})} \geq 2^{-n}\sqrt{kf(\boldsymbol{z}_0)} = r. \tag{A.16}$$

Setting $K = 2\frac{\eta}{r}$, we can apply Theorem 17 to get the existence of a near-steepest descent curve $\boldsymbol{z}(t) : [0, L] \to \mathbb{R}^n$ with $\boldsymbol{z}(0) = \boldsymbol{z}_0$; $f(\boldsymbol{z}(L)) \leq \frac{f(\boldsymbol{z}_0)}{2^{2n+1}}$; and

$$\frac{\mathrm{d}}{\mathrm{d}t} f(\boldsymbol{z}(t)) \leq -\sqrt{2kf(\boldsymbol{z}(t))}\,. \tag{A.17}$$

By inequality with the differential equation $\frac{\mathrm{d}y}{\mathrm{d}t} = -2\sqrt{\frac{ky}{2}}$, we have that

$$f(\boldsymbol{z}(t)) \leq \left(\sqrt{f(\boldsymbol{z}_0)} - \sqrt{\frac{k}{2}}t\right)^2, \tag{A.18}$$

on $t \in [0, L]$. Solving the above, we find that the first t at which $f($
$X(t))$ drops below $\frac{f(\boldsymbol{z}_0)}{2^{2n+1}}$ obeys

$$t \leq \sqrt{\frac{2f(\boldsymbol{z}_0)}{k}}\,. \tag{A.19}$$

Thus as $\boldsymbol{z}(t)$ is 1-Lipschitz, there exists a corresponding $\boldsymbol{z}_n = \boldsymbol{z}(t)$ in a ball of radius $\sqrt{\frac{2f(\boldsymbol{z}_0)}{k}}$ with $f(\boldsymbol{z}_n) \leq \frac{f(\boldsymbol{z}_0)}{2^{2n+1}}$. As f is continuous, dropping to a convergent subsequence of $\boldsymbol{z}_n$ we determine that there exists $\boldsymbol{z}_\infty = \lim_{n\to\infty} \boldsymbol{z}_n$ with

- $f(\boldsymbol{z}_\infty) = 0$.

- $\|\boldsymbol{z}_0 - \boldsymbol{z}_\infty\|_2 \leq \sqrt{\frac{2f(\boldsymbol{z}_0)}{k}}$.

Thus $f(\boldsymbol{z}_0) \geq \frac{k}{2}\|\boldsymbol{z}_0 - \boldsymbol{z}_\infty\|_2^2$ and f is k-QG. □

A.4. Bilevel Optimization

This thesis makes frequent used of nested or bilevel optimization problems. We now describe some theoretical tools for analysis of problems of this type, including variants of the Envelope theorem. We consider the following domain of convex optimization problems: Let $f(\boldsymbol{z}) = \min_{\boldsymbol{\Lambda}\in\mathcal{K}} g(\boldsymbol{z}, \boldsymbol{\Lambda})$, $\mathcal{K}^*(\boldsymbol{z}) = \arg\min_{\boldsymbol{\Lambda}\in\mathcal{K}} g(\boldsymbol{z}, \boldsymbol{\Lambda})$ with with the following assumptions:

1. $\mathcal{K} \subseteq \mathbb{R}^m$ is closed and convex.
2. g is convex in $\boldsymbol{z}$ at any $\boldsymbol{\Lambda} \in \mathcal{K}$
3. g is convex in $\boldsymbol{\Lambda}$ at any $\boldsymbol{z}$
4. g is jointly semiconvex in $(\boldsymbol{z}, \boldsymbol{\Lambda})$ on $\mathbb{R}^n \times \mathbb{R}^m$
5. g is Lipschitz on any compact domain.
6. g has a finite minimizer (and therefore $\mathcal{K}^*(\boldsymbol{z})$ is closed and nonempty).
7. g has continuous norm bound $\alpha(\boldsymbol{z})$ on the minimizers $\mathcal{K}^*(\boldsymbol{z})$:

$$\max_{\boldsymbol{\Lambda}\in\mathcal{K}^*(x)} \|\boldsymbol{\Lambda}\|_2 \leq \alpha(\boldsymbol{z})\,.$$

These properties have some immediate implications. As g is jointly semiconvex, it is also subdifferentiable, and $\partial g(\boldsymbol{z}, \boldsymbol{\Lambda})$ is u.s.c. in $(\boldsymbol{z}, \boldsymbol{\Lambda})$. As g is also locally Lipschitz, $\partial g(\boldsymbol{z}, \boldsymbol{\Lambda})$ is locally bounded. These properties also imply a semi-continuity condition on the strong slope of g:

Lemma 13. $\|\nabla\|_{\boldsymbol{z}}\, g(\boldsymbol{z}, \boldsymbol{\Lambda})$ *is lower semi-continuous in* $\boldsymbol{z}$*, i.e.*

$$\|\nabla\|_{\boldsymbol{z}}\, g(\boldsymbol{z}, \boldsymbol{\Lambda}) \leq \|\nabla\|_{\boldsymbol{z}} \lim_{\boldsymbol{y}\to\boldsymbol{z}} g(\boldsymbol{z})\,. \tag{A.20}$$

Proof. Follows directly from the local boundedness and upper semicontinuity of $\partial g(\boldsymbol{z}, \boldsymbol{\Lambda})$ and the equality

$$\|\nabla\|_{\boldsymbol{z}} g(\boldsymbol{z}, \boldsymbol{\Lambda}) = \min_{\boldsymbol{g}^* \in \partial_{\boldsymbol{z}} \boldsymbol{g}(\boldsymbol{z}, \boldsymbol{\Lambda})} \|\boldsymbol{g}^*\| \ .$$

□

We now show in detail that f inherits several properties from g. First, we show that the f inherits continuity from g and thus the strong slope of f is well-defined:

Lemma 14. *The restriction of f to any compact domain is Lipschitz.*

Proof. It suffices to check the condition on any centered norm ball. Let $\boldsymbol{z} \in \mathbb{R}^n$. Let $\varepsilon > 0$. Let $\boldsymbol{\Lambda}_{max} = \max_{\|\bar{\boldsymbol{z}}\|_2 \leq \|\boldsymbol{z}\|_2 + \varepsilon} \alpha(\bar{\boldsymbol{z}}) < \infty$. Let L_g be the Lipschitz constant of g on $\{\bar{\boldsymbol{z}} : \|\boldsymbol{z} - \bar{\boldsymbol{z}}\|_2 \leq \varepsilon\} \times \{\|\boldsymbol{\Lambda}\|_2 \leq \boldsymbol{\Lambda}_{max}\} \cap \mathcal{K}$. Let $\|\boldsymbol{z} - \boldsymbol{z}'\|_2 \leq \varepsilon$ and let $\boldsymbol{\Lambda}, \boldsymbol{\Lambda}'$ be solutions to the underlying minimizations in $f(\boldsymbol{z}), f(\boldsymbol{z}')$. Thus,

$$\begin{aligned}
-L_g \left\|\boldsymbol{z} - \boldsymbol{z}'\right\|_2 &\leq g(\boldsymbol{z}, \boldsymbol{\Lambda}) - g(\boldsymbol{z}', \boldsymbol{\Lambda}) \,, && \text{(A.21)} \\
&\leq g(\boldsymbol{z}, \boldsymbol{\Lambda}) - g(\boldsymbol{z}', \boldsymbol{\Lambda}') \,, && \text{(A.22)} \\
&= f(\boldsymbol{z}) - f(\boldsymbol{z}') \,, && \text{(A.23)} \\
&\leq g(\boldsymbol{z}, \boldsymbol{\Lambda}') - g(\boldsymbol{z}', \boldsymbol{\Lambda}') \,, && \text{(A.24)} \\
&\leq L_g \left\|\boldsymbol{z} - \boldsymbol{z}'\right\|_2 \ . && \text{(A.25)}
\end{aligned}$$

Thus f is Lipschitz on $\{\bar{\boldsymbol{z}} : \|\boldsymbol{z} - \bar{\boldsymbol{z}}\|_2 \leq \varepsilon\}$ with constant L_g. □

The solution set instead inherits semicontinuity:

Lemma 15. *$\mathcal{K}^*(\boldsymbol{z})$ is upper semicontinuous.*

Proof. As the range of $\mathcal{K}^*(\boldsymbol{z})$ is compact over any compact domain it suffices to show that for any $\boldsymbol{\Lambda}_i \to \boldsymbol{\Lambda}$, $\boldsymbol{z}_i \to \boldsymbol{z}$, $\boldsymbol{\Lambda}_i \in \mathcal{K}^*(\boldsymbol{z}_i)$, that $\boldsymbol{\Lambda} \in \mathcal{K}^*(\boldsymbol{z})$.

$\boldsymbol{\Lambda} \in \mathcal{K}$ as $\boldsymbol{\Lambda}$ is closed. Let $\bar{\boldsymbol{\Lambda}} \in \mathcal{K}^*(\boldsymbol{z})$. Then $g(\boldsymbol{z}_i, \bar{\boldsymbol{\Lambda}}) \to g(\boldsymbol{z}, \bar{\boldsymbol{\Lambda}}) = f(\boldsymbol{z})$ by local continuity of g. But $g(\boldsymbol{z}_i, \bar{\boldsymbol{\Lambda}}) \geq g(\boldsymbol{z}_i, \boldsymbol{\Lambda}_i)$ as $\boldsymbol{\Lambda}_i \in \mathcal{K}^*(\boldsymbol{z}_i)$. Thus $f(\boldsymbol{z}) \geq g(\boldsymbol{z}, \boldsymbol{\Lambda})$ and by definition of $\mathcal{K}^*$ and f, $\boldsymbol{\Lambda} \in \mathcal{K}^*(\boldsymbol{z})$. □

Envelope theorems

In its simplest form, the Envelope theorem (Afriat, 1971) states that for smooth f, g, $\nabla f = \nabla g|_{\boldsymbol{\Lambda} \in \mathcal{K}^*}$—in other words, f inherits its gradient of g. We will deal with non-smooth optimization in Chapter 5, and thus we extend this concept from gradients to strong slopes:

Theorem 19 (Strong Slope Envelope Theorem)**.**

$$\|\nabla\| f(\boldsymbol{z}) \geq \max_{\boldsymbol{\Lambda} \in \mathcal{K}^*(\boldsymbol{z})} \|\nabla\|_{\boldsymbol{z}} g(\boldsymbol{z}, \boldsymbol{\Lambda}). \tag{A.26}$$

Proof. Let $\boldsymbol{\Lambda} \in \mathcal{K}^*(\boldsymbol{z})$. $\|\nabla\| f$ is well-defined due to its local Lipschitz property. $\|\nabla\|_{\boldsymbol{z}} g(\boldsymbol{z}, \boldsymbol{\Lambda})$ is also well-defined due to its continuity. By the optimality of the minimization in f, for any $\boldsymbol{y}$,

$$\frac{f(\boldsymbol{y}) - f(\boldsymbol{z})}{\|\boldsymbol{y} - \boldsymbol{z}\|} \leq \frac{g(\boldsymbol{y}, \boldsymbol{\Lambda}) - g(\boldsymbol{z}, \boldsymbol{\Lambda})}{\|\boldsymbol{y} - \boldsymbol{z}\|}. \tag{A.27}$$

Thus as neg is monotonically decreasing,

$$\frac{\operatorname{neg}(f(\boldsymbol{y}) - f(\boldsymbol{z}))}{\|\boldsymbol{y} - \boldsymbol{z}\|} \geq \frac{\operatorname{neg}(g(\boldsymbol{y}, \boldsymbol{\Lambda}) - g(\boldsymbol{z}, \boldsymbol{\Lambda}))}{\|\boldsymbol{y} - \boldsymbol{z}\|}. \tag{A.28}$$

The claim follows from taking the appropriate lim sup of the above expression and using the definition of strong slope. □

Corollary 3 (Limiting Slope Envelope Theorem)**.**

$$\overline{\|\nabla\|} f(\boldsymbol{z}) \geq \min_{\boldsymbol{\Lambda} \in \mathcal{K}^*(\boldsymbol{z})} \|\nabla\|_{\boldsymbol{z}} g(\boldsymbol{z}, \boldsymbol{\Lambda}).$$

Proof. Let $\boldsymbol{z}_n \to \boldsymbol{z}$. Then we may construct a convergent $\boldsymbol{\Lambda}(\boldsymbol{z}_n) \in \mathcal{K}^*(\boldsymbol{z}_n)$, and $\boldsymbol{\Lambda}(\boldsymbol{z}_n) \to \boldsymbol{\Lambda} \in \mathcal{K}^*(\boldsymbol{z})$ by upper semi-continuity of $\mathcal{K}^*(\boldsymbol{z})$ (Lemma 15). Then, $\|\nabla\| f(\boldsymbol{y}_n) \geq \|\nabla\|_{\boldsymbol{z}} g(\boldsymbol{z}_n, \boldsymbol{\Lambda}(\boldsymbol{z}_n))$ by strong

slope envelope dominance. $\|\nabla\|_{\boldsymbol{z}} g(\boldsymbol{z}_n, \boldsymbol{\Lambda}(\boldsymbol{z}_n))$ converges to $g^* \geq \|\nabla\|_{\boldsymbol{z}} g(\boldsymbol{z}, \boldsymbol{\Lambda})$ by Lemma 13. The claim follows as $\|\nabla\| f(\boldsymbol{z}_n)$ is bounded below in the limit by g^*. □

Descent Dominance

We finally note that f inherits descent dominance properties from g:

Corollary 4 (Inherited Descent Dominance)**.** *Assume that* $\min_{\boldsymbol{z}} f(\boldsymbol{z}) = 0$*. Then* f *is* k*-DD if*

$$\min_{\boldsymbol{\Lambda} \in \mathcal{K}^*(\boldsymbol{z})} \|\nabla\|_{\boldsymbol{z}} g(\boldsymbol{z}, \boldsymbol{\Lambda}) = \min_{\boldsymbol{\Lambda} \in \mathcal{K}^*(\boldsymbol{z})} \min_{\boldsymbol{g}^* \in \partial_{\boldsymbol{z}} g(\boldsymbol{z}, \boldsymbol{\Lambda})} \|\boldsymbol{g}^*\|_2 \geq \sqrt{2k g(\boldsymbol{z}, \mathcal{K}^*(\boldsymbol{z}))}\,. \tag{A.29}$$

Proof. Follows directly from Corollary 3 and the semicontinuity of g. □

The result of the above shows that f is k-DD and therefore k-QG as long as g has large partial subgradients in $\boldsymbol{z}$ at any solution $\boldsymbol{\Lambda}^*$.

APPENDIX B

ADDITIONAL EXPERIMENTAL DETAILS FOR CHAPTER 4

B.1. 1D Example

We now detail the prediction task explored in Figure 4.1. The 1D system has state $x = [z;\ \dot{z}]$ and continuous-time dynamics

$$\ddot{z} = \begin{cases} -9.81 & z > 0\,, \\ -kz - 2\sqrt{k}\dot{z} - 9.81 & z \leq 0\,. \end{cases} \tag{B.1}$$

For each stiffness setting, data $(\dot{z}_t, \dot{z}_{t+1})$ are selected with an initial velocity $\dot{z}_t \sim \mathcal{U}([-3, 5])$, and simulated (B.1) with initial condition $[1;\ \dot{z}_t]$ for 1 s to generate the final velocity $\dot{z}_{t+1}$. We add gaussian noise ($\sigma^2 = .01$) to both velocities.

For each stiffness, we train 100 models on different sets of 20 training and 20 validation datapoints selected from this distribution; one such training set is displayed in Figure 4.1b. Each model is an MLP with input $\dot{z}_t$, two hidden layers of width 128, and output $\dot{z}_{t+1}$. Models are trained with prediction error loss, and terminated with early stopping with a patience of 10 epochs. The Adam optimizer is used, with learning rate and weight decay separately tuned for each stiffness by grid search on $\{1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}\}$ and $\{1 \times 10^{-2}, 1 \times 10^{-4}, 0\}$, respectively, to minimize ground-truth prediction error. The average prediction of these models with a 1 std. dev. window are plotted in Figure 4.1b.

B.2. Simulation Details

Here, we provide additional details on the physics of MuJoCo and the data generation process.

B.2.1. Interpenetration in MuJoCo

Here we discussion interpenetration $\boldsymbol{\phi}(\boldsymbol{q})$ in MuJoCo. During contact, $\boldsymbol{\phi} = 0$ may be considered a constraint which must be stabilized for reliable simulation; the seminal approach of Baumgarte Baumgarte (1972) is to enforce penetration to obey dynamics inspired by a spring-mass damper: $\ddot{\boldsymbol{\phi}}_B = -(k\boldsymbol{\phi} + b\dot{\boldsymbol{\phi}})$. To scale this idea to efficient multibody simulation, MuJoCo computes a convex

optimization-based approximation Todorov (2014):

$$\ddot{\boldsymbol{\phi}} \approx (1 - d(\boldsymbol{\phi}))\ddot{\boldsymbol{\phi}}_s + d(\boldsymbol{\phi})\ddot{\boldsymbol{\phi}}_B \,, \tag{B.2}$$

where $\ddot{\boldsymbol{\phi}}_s$ is the acceleration due to gravity and inertia (-9.81 for the point mass in Figure 4.1), and $d(\boldsymbol{\phi}) \approx 1$ is a user-specified function. A thorough description of this function is available in the MuJoCo documentation online [8].

B.2.2. Data Generation

To generate an initial state, we generate a perturbation Δx_0 around the nominal state $x_{0,ref}$, an initial condition taken from the *ContactNets* dataset (Pfrommer* et al., 2020):

$$\begin{aligned}\boldsymbol{x}_{0,ref} = [&0.186,\ 0.026,\ 0.122,\ -0.525,\ 0.394,\ -0.296,\\ &-0.678,\ 0.014,\ 1.291,\ -0.212,\ 1.463,\ -4.854,\ 9.870]\,,\end{aligned}$$

where the cube center of mass is ~ 0.12m above the ground, with initial downward velocity of $\sim 0.2\ \frac{\text{m}}{\text{s}}$. This perturbation consists of $\Delta \boldsymbol{p}_0 \sim \mathcal{U}([-0.1, 0.1]^3)\ \frac{\text{m}}{\text{s}}$; $\Delta \boldsymbol{q}_0 \sim Q\left(\theta \frac{\boldsymbol{r}}{||\boldsymbol{r}||_2}\right)$, a body-axis rotation of angle $\theta \sim \mathcal{U}([-1, 1])$ rad and axis $\boldsymbol{r} \sim \mathcal{U}([1, 1]^3)$; $\Delta \dot{\boldsymbol{p}}_0 \sim \mathcal{U}([-0.1, 0.1]^3)\ \frac{\text{m}}{\text{s}}$; and $\Delta \boldsymbol{\omega}_0 \sim \mathcal{U}([-0.1, 0.1]^3)\ \frac{\text{rad}}{\text{s}}$. Since the average trajectory length for the Hard setting is 80 time-steps ($\sim$ 540ms), we truncate the *Medium* and *Soft* trajectories to 80 time-steps to make the amount of data per trajectory equal across all settings. After a trajectory is generated from this initial condition, we add two forms of noise to the configurations $[\boldsymbol{p}_t;\ \boldsymbol{q}_t]$. First we add constant-in-time noise to represent cumulative sensor drift: $\Delta \boldsymbol{p} \sim \mathcal{U}([1, 1]^3)$ mm and $\Delta \boldsymbol{q} \sim Q\left(\theta \frac{\boldsymbol{r}}{||\boldsymbol{r}||_2}\right)$, with $\theta \sim \mathcal{U}([-1, 1])$ deg and axis $\boldsymbol{r} \sim \mathcal{U}([1, 1]^3)$. Then, we add i.i.d. noise to each measurement to model the small inconsistencies between measurements in the same fashion: $\Delta \boldsymbol{p}_t \sim \mathcal{U}([.01, .01]^3)$ mm and $\Delta \boldsymbol{q}_t \sim Q\left(\theta \frac{\boldsymbol{r}}{||\boldsymbol{r}||_2}\right)$, with $\theta \sim \mathcal{U}([-.01, .01])$ deg and axis $\boldsymbol{r} \sim \mathcal{U}([1, 1]^3)$. Velocities are reconstructed from the noisy data by inverting the finite difference equations (4.6)–(4.7). While measurements have very small relative position noise, finite differencing amplifies the velocity noise

[8] http://www.mujoco.org/book/modeling.html

Table B.1: Hyperparameter Search Space

Hyperparameter	Values		
target variable	v_{t+1}	Δv	
learning-rate	1e-3	1e-4	1e-5
hidden-size	128	256	512
history-length	4	8	16
weight-decay	0	4e-5	4e-3

by $\frac{1}{\Delta t} \approx 160\text{s}^{-1}$.

B.3. Learning Details

Our MLPs consist of 4 hidden fully-connected layers with ReLU activations, plus a final linear layer. Our RNNs maintain a hidden state $\boldsymbol{z}_t$ with initial value $\mathbf{0}$. $\boldsymbol{z}_t$ is updated sequentially for each $\boldsymbol{x}_t$ as $\boldsymbol{z}_t = \boldsymbol{\psi}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{z}_{t-1})$ from the previous hidden-state $\boldsymbol{z}_{t-1}$, where $\boldsymbol{\psi}_{\boldsymbol{\theta}}$ is a learned non-linear function. By recursively unfolding (visualized in Fig. B.1),

$$z_t = \boldsymbol{\psi}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{\psi}_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}, \dots \boldsymbol{\psi}_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-h+1}, \mathbf{0}) \dots)) \,. \tag{B.3}$$

Finally, we use a two layer fully-connected network as a decoder $\boldsymbol{\psi}_{\boldsymbol{\theta},dec}$ to extract the predicted velocity vector as $\boldsymbol{v}_{t+1} = \boldsymbol{\psi}_{\boldsymbol{\theta},dec}(z_t)$. The decoder consists of a hidden layer of width half the size of RNN hidden-state followed by ReLU activation units and the output layer. For each MLP and RNN architecture, we tried different target variables and sweep over different values of learning-rate, hidden-layer size, and weight-decay, centered around hand-tuned values. While the history-length is 1 for MLPs, for RNNs we also tried different history-lengths. Table B.1 provides the space of hyperparameters sweeped over in this process. For each combination of settings, we complete at least 10 training runs on high data regime of 500 example trajectories, and then select the setting with the lowest average prediction error over $\mathcal{D}_{\text{test}}$.

While MLP's and RNN's had similar training error and generalization error trends, the long-term prediction error was noticeably different (see Fig. B.2). To make a fair comparison with our RNN's of history-length $h = 16$, our MLP rollout experiments also start from the 16th time-step. RNN's are worse than MLP's on *Hard* model rollouts, in spite of their superior performance on single-step

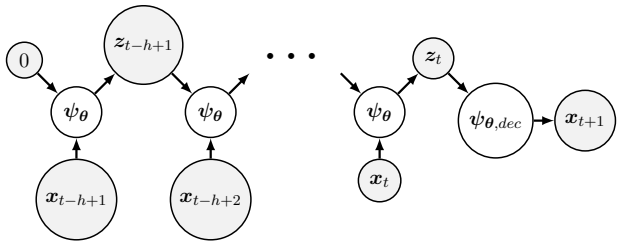


Figure B.1: The structure of our RNN predictors. ψ_{θ} is a recurrent unit (GRU), while $\psi_{\theta,dec}$ is an MLP decoder.

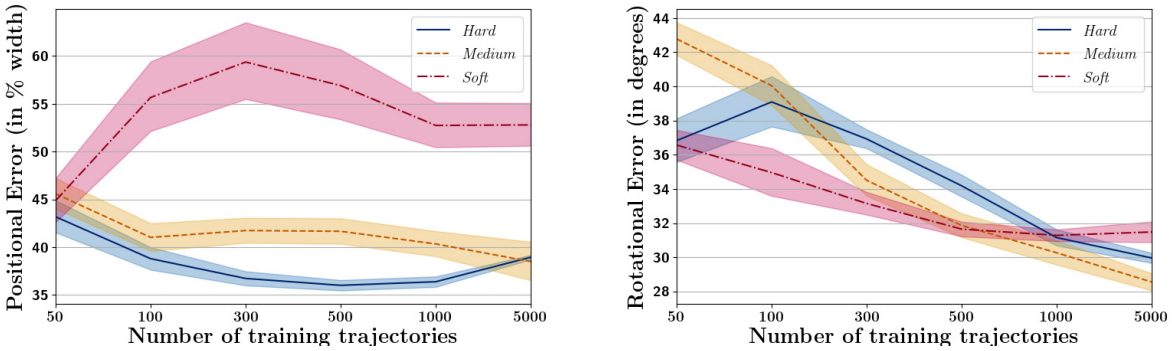


Figure B.2: We plot the performance of our optimized MLP networks on long-term prediction of position and orientation with 95% confidence intervals.

predictions. We speculate that RNN models can exploit temporal consistency between velocities for smooth systems. After all, if the trajectory is smooth in time, then one can predict the velocity over a short horizon by simply doing polynomial extrapolation. However, as the velocity is nearly discontinuous in time for hard models, this intuition may no longer be valid.

APPENDIX C

PROOFS FOR CHAPTER 5

C.1. Proof of Lemma 2

As both $\mathcal{L}_{\text{prediction}}$ (2.49) and $\mathcal{L}_{\text{graph}}$ (5.4) are averaged over the datapoints in $\mathcal{D}$, W.L.O.G. we may consider a dataset consisting of a single datapoint $(\boldsymbol{x}_i, \boldsymbol{u}_i, \boldsymbol{v}_i')$. As $[\boldsymbol{x}_i, \boldsymbol{u}_i, \boldsymbol{f_\theta}(\boldsymbol{x}_i, \boldsymbol{u}_i)] \in \text{Graph}(\boldsymbol{f_\theta})$, the first inequality follows directly:

$$\mathcal{L}_{\text{prediction}}(\boldsymbol{f_\theta}, \mathcal{D}) = \left\| \begin{bmatrix} \mathbf{0} \\ \boldsymbol{v}_i' - \boldsymbol{f_\theta}(\boldsymbol{x}_i, \boldsymbol{u}_i) \end{bmatrix} \right\|_2^2 \geq \min_{\bar{\boldsymbol{x}}_i} \left\| \begin{bmatrix} \bar{\boldsymbol{x}}_i - \boldsymbol{x}_i \\ \boldsymbol{v}_i' - \boldsymbol{f_\theta}(\bar{\boldsymbol{x}}_i, \boldsymbol{u}_i) \end{bmatrix} \right\|_2^2 = \mathcal{L}_{\text{graph}}(\boldsymbol{f_\theta}, \mathcal{D}) \,. \tag{C.1}$$

To prove the second inequality, we first apply the Lipschitz condition of $\boldsymbol{f_\theta}$ at $\boldsymbol{x}_i$ to give the following:

$$\text{Graph}(\boldsymbol{f_\theta}(\cdot, \boldsymbol{u}_i, \cdot)) \subseteq \mathcal{G} = \left\{ \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{v} \end{bmatrix} : \|\boldsymbol{v} - \boldsymbol{f_\theta}(\boldsymbol{x}_i, \boldsymbol{u}_i)\|_2 - L_f \|\boldsymbol{x} - \boldsymbol{x}_i\|_2 \leq 0 \right\} . \tag{C.2}$$

We can therefore lower bound $\mathcal{L}_{\text{graph}}$ as

$$\mathcal{L}_{\text{graph}}(\boldsymbol{x}_i, \boldsymbol{v}_i) \geq \min_{[\boldsymbol{x};\, \boldsymbol{v}'] \in \mathcal{G}} \left\| \begin{bmatrix} \boldsymbol{x} - \boldsymbol{x}_i \\ \boldsymbol{v}' - \boldsymbol{v}_i' \end{bmatrix} \right\|_2^2 . \tag{C.3}$$

We can write the KKT conditions of (C.3) with Lagrange multiplier $\gamma \geq 0$ as

$$0 = \begin{bmatrix} \boldsymbol{x} - \boldsymbol{x}_i \\ \boldsymbol{v}' - \boldsymbol{v}_i' \end{bmatrix} + \gamma \begin{bmatrix} -L_f \frac{\boldsymbol{x} - \boldsymbol{x}_i}{\|\boldsymbol{x} - \boldsymbol{x}_i\|_2} \\ \frac{\boldsymbol{v}' - \boldsymbol{f_\theta}(\boldsymbol{x}_i, \boldsymbol{u}_i)}{\|\boldsymbol{v}' - \boldsymbol{f_\theta}(\boldsymbol{x}_i, \boldsymbol{u}_i)\|_2} \end{bmatrix} , \tag{C.4}$$

$$0 \geq \left\|\boldsymbol{v}' - \boldsymbol{f_\theta}(\boldsymbol{x}_i, \boldsymbol{u}_i)\right\|_2 - L_f \|\boldsymbol{x} - \boldsymbol{x}_i\|_2 \,, \tag{C.5}$$

$$0 = \gamma \left(\left\|\boldsymbol{v}' - \boldsymbol{f_\theta}(\boldsymbol{x}_i, \boldsymbol{u}_i)\right\|_2 - L_f \|\boldsymbol{x} - \boldsymbol{x}_i\|_2 \right) . \tag{C.6}$$

When $\boldsymbol{v}' \neq \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{u}_i)$, these conditions are solved by

$$\boldsymbol{x}^* = \boldsymbol{x}_i + \hat{\boldsymbol{r}} \frac{L_f}{1 + L_f^2} \left\| \boldsymbol{v}' - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{u}_i) \right\|_2 , \tag{C.7}$$

$$\boldsymbol{v}^{*\prime} = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{u}_i) + L_f \left\| \boldsymbol{x}^* - \boldsymbol{x}_i \right\|_2 \frac{\boldsymbol{v}_i' - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{u}_i)}{\left\| \boldsymbol{v}_i' - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{u}_i) \right\|_2} , \tag{C.8}$$

$$\gamma = \frac{\left\| \boldsymbol{x}^* - \boldsymbol{x}_i \right\|_2}{L_f} , \tag{C.9}$$

where $\hat{\boldsymbol{r}}$ is an arbitrary unit direction. Substituting the formulas for $\boldsymbol{x}^*$ and $\boldsymbol{v}^*$ into (C.3) yields the final result.

C.2. Simplifications, lemmas, formulas and additional notation supporting Theorem 1

Without loss of generality, by a change of units in $\boldsymbol{\Lambda}_{t,i}$ from [N] to [$\boldsymbol{\mu}_i$N], we may assume $\boldsymbol{\mu}_i = 1$, and that the feasible set in the loss is

$$\mathcal{K} = L^m , \tag{C.10}$$

$$L = \{ \boldsymbol{\Lambda} : \boldsymbol{\Lambda}_n \geq \left\| \boldsymbol{\Lambda}_t \right\|_2 \} . \tag{C.11}$$

L is called the *Lorentz cone*, a convex cone. It is self-dual, such that $L = \{ \boldsymbol{\Lambda} : \boldsymbol{\Lambda} \cdot L \geq 0 \}$, i.e. $L = \text{dual}(L)$. Under the simplifying assumptions in Theorem 1,

$$\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\Lambda}) = \boldsymbol{v} + \boldsymbol{\tau} + \boldsymbol{J}^T \boldsymbol{\Lambda} , \tag{C.12}$$

$$\boldsymbol{\phi}(\boldsymbol{q}) = \boldsymbol{c} + \boldsymbol{J}_n \boldsymbol{q} , \tag{C.13}$$

where $\boldsymbol{\tau} = F_s \Delta t$ and $\boldsymbol{c}$ are constants.

In this section, we will consider evaluation of the $\mathcal{L}_{\text{violation}}$ over single datapoint $\mathcal{D} = \{(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}')\}$. Let $\boldsymbol{\Lambda}^*$ be a solution to the minimization in (5.5). As L and therefore $\mathcal{K}$ are self-dual convex cones,

the KKT conditions on $\boldsymbol{\Lambda}^* \in \mathcal{K}^*(\boldsymbol{x}, \boldsymbol{v}')$ are

$$L \ni \boldsymbol{\Lambda}_i^* \perp \boldsymbol{c}_i + \boldsymbol{d}_i - \boldsymbol{J}_i \tilde{\boldsymbol{v}} \in L\,, \tag{C.14}$$

$$\tilde{\boldsymbol{v}} = \boldsymbol{v} - \boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\Lambda}^*) = \boldsymbol{v}' - \left(\boldsymbol{v} + \boldsymbol{\tau} + \boldsymbol{J}^T \boldsymbol{\Lambda}^*\right)\,. \tag{C.15}$$

Define also the lumped terms

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{c}_1 \\ \vdots \\ \boldsymbol{c}_m \end{bmatrix}\,, \qquad \boldsymbol{d} = \begin{bmatrix} \boldsymbol{d}_1 \\ \vdots \\ \boldsymbol{d}_m \end{bmatrix}\,. \tag{C.16}$$

C.2.1. Problem Class

We now show that the loss is in the problem class of bilevel problems explored in Section A.4, with $\boldsymbol{z} \sim [\boldsymbol{x}, \boldsymbol{v}']$, $f \sim \mathcal{L}_{\text{violation}}(\boldsymbol{f}_{\boldsymbol{\theta}}, \{\cdot, \boldsymbol{u}, \cdot\})$ and $g \sim l_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}', \boldsymbol{\Lambda})$:

1. $\mathcal{K} = L^m$ is the Cartesian product of closed Lorentz cones, and therefore is closed.

2. $\frac{1}{2}\left\|\boldsymbol{v}' - \boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\Lambda})\right\|_2^2$ is jointly convex quadratic in $\boldsymbol{x}, \boldsymbol{v}', \boldsymbol{\Lambda}$ from the above equations. As $\bar{\boldsymbol{\phi}}(\boldsymbol{q}, \boldsymbol{v}')$ is affine in $(\boldsymbol{q}, \boldsymbol{v}')$ (2.36); $\text{pos}(\cdot)$, $\|\cdot\|_2$, and $\text{neg}(\cdot)$ are convex; and $\boldsymbol{\Lambda}_n \geq 0$ inside $\mathcal{K}$, $h_{\boldsymbol{\theta}}$ is also convex in $\boldsymbol{x}, \boldsymbol{v}'$ on $\boldsymbol{\Lambda}$. Thus, $l_{\boldsymbol{\theta}}$ is convex in $\boldsymbol{x}, \boldsymbol{v}'$ on $\boldsymbol{\Lambda}$.

3. $h_{\boldsymbol{\theta}}$ is affine in $\boldsymbol{\Lambda}$ and therefore convex in $\boldsymbol{\Lambda}$; thus $l_{\boldsymbol{\theta}}$ is also convex in $\boldsymbol{\Lambda}$ at all $\boldsymbol{x}, \boldsymbol{v}'$.

4. We can extend $h_{\boldsymbol{\theta}}$ (and thus $l_{\boldsymbol{\theta}}$) outside of $\mathbb{R}^n \times \mathcal{K}$ as

$$\hat{h_{\boldsymbol{\theta}}} = \sum_i \frac{1}{2}\text{neg}(\bar{\boldsymbol{\phi}}_i)^2 + \boldsymbol{c}_i \cdot \bar{\boldsymbol{\Lambda}}(\boldsymbol{\Lambda}_i) + \boldsymbol{d}_i \cdot \bar{\boldsymbol{\Lambda}}(\boldsymbol{\Lambda}_i)\,, \tag{C.17}$$

$$\boldsymbol{\Lambda}(\boldsymbol{\Lambda}_i) = [\text{pos}(\boldsymbol{\Lambda}_{n,i}); \boldsymbol{\Lambda}_{t,i}]\,. \tag{C.18}$$

   $\boldsymbol{\Lambda}(\boldsymbol{\Lambda}_i) \cdot (\boldsymbol{c}_i + \boldsymbol{d}_i)$ and $\text{neg}(\bar{\boldsymbol{\phi}}_i)^2$ can easily be shown to be convex in $\boldsymbol{y}_i = [\boldsymbol{\Lambda}_i; \bar{\boldsymbol{\phi}}_i; \boldsymbol{J}_t \boldsymbol{v}']$ up to $4\left\|\boldsymbol{y}_i\right\|_2^2$. As $\boldsymbol{y}$ is affine in $(\boldsymbol{x}, \boldsymbol{v}', \boldsymbol{\Lambda})$, $h_{\boldsymbol{\theta}}$ and therefore $l_{\boldsymbol{\theta}}$ are jointly semiconvex in $(\boldsymbol{x}, \boldsymbol{v}', \boldsymbol{\Lambda})$.

5. $l_{\boldsymbol{\theta}}$ is jointly quadratic in $\left\{\boldsymbol{y}_i = [\boldsymbol{\Lambda}_i; \mathrm{pos}(\bar{\boldsymbol{\phi}}_i); \mathrm{neg}(\bar{\boldsymbol{\phi}}_i); \boldsymbol{v}; \boldsymbol{v}'; \|\boldsymbol{J}_{t,i}\boldsymbol{v}'\|_2]\right\}$, which is Lipschitz in $\boldsymbol{x}, \boldsymbol{v}', \boldsymbol{\Lambda}$, and thus $l_{\boldsymbol{\theta}}$ is Lipschitz on any compact domain.

6. As the loss is a convex SOCP with an objective that grows unboundedly as $\|\boldsymbol{\Lambda}\| \to \infty$ (see below), it has a finite minimizer.

7. As the friction cone is pointed (Assumption 2), $\left\|\boldsymbol{J}^T\boldsymbol{\Lambda}\right\|_2 \geq p\,\|\boldsymbol{\Lambda}\|_2$ for $p > 0$ any $\boldsymbol{\Lambda} \in \boldsymbol{\Lambda}$. From the KKT we have

$$
\begin{aligned}
0 &\geq \boldsymbol{\Lambda}^* \cdot (\boldsymbol{c} + \boldsymbol{d} - \boldsymbol{J}\tilde{\boldsymbol{v}})\,, && \text{(C.19)}\\
&\geq -\|\boldsymbol{\Lambda}^*\|_2 \left\|\boldsymbol{c} + \boldsymbol{d} - J(\boldsymbol{v}' - v - \boldsymbol{\tau})\right\|_2 + p^2\,\|\boldsymbol{\Lambda}^*\|_2^2\,, && \text{(C.20)}\\
&\geq \|\boldsymbol{\Lambda}^*\|_2 \left(-\left\|\boldsymbol{c} + \boldsymbol{d} - J(\boldsymbol{v}' - v - \boldsymbol{\tau})\right\|_2 + p^2\,\|\boldsymbol{\Lambda}^*\|_2\right). && \text{(C.21)}
\end{aligned}
$$

Thus we may pick

$$
\alpha(\boldsymbol{x}, \boldsymbol{v}') = \frac{1}{p^2}\left\|\boldsymbol{c} + \boldsymbol{d} - J(\boldsymbol{v}' - v - \boldsymbol{\tau})\right\|_2\,. \tag{C.22}
$$

C.2.2. Lemmas

Lemma 16. *Let $\boldsymbol{\Lambda}^*$ be a minimizer of $l_{\boldsymbol{\theta}}$ at $\boldsymbol{x}, \boldsymbol{v}'$. Let $\hat{\boldsymbol{t}}_i \in \mathrm{Unit}(\boldsymbol{J}_{t,i}\boldsymbol{v}')$ and $\boldsymbol{r}_i = [-1; \hat{\boldsymbol{t}}_i]$. Then,*

$$
\boldsymbol{\Lambda}_i^* \cdot \boldsymbol{d}_i \leq \partial_{\boldsymbol{v}'}\left(\boldsymbol{\Lambda}_i^* \cdot \boldsymbol{d}_i\right) \cdot \tilde{\boldsymbol{v}}\,. \tag{C.23}
$$

Proof. As $\boldsymbol{\Lambda}_{n,i}^* \geq 0$, we have by chain rule that

$$
\partial_{\boldsymbol{v}'}\left(\boldsymbol{\Lambda}_i^* \cdot \boldsymbol{d}_i\right) \cdot \tilde{\boldsymbol{v}} = \boldsymbol{\Lambda}_i^{*T} \begin{bmatrix} \mathrm{Unit}(\boldsymbol{J}_{t,i}\boldsymbol{v}')^T \\ \boldsymbol{I}_2 \end{bmatrix} \boldsymbol{J}_{t,i}\tilde{\boldsymbol{v}}\,. \tag{C.24}
$$

Pick an element $\hat{\boldsymbol{t}}_i$ of $\mathrm{Unit}(\boldsymbol{J}_{t,i}\boldsymbol{v}')$. Let

$$
\boldsymbol{r}_i = \begin{bmatrix} -1 \\ \hat{\boldsymbol{t}}_i \end{bmatrix} \in -L\,. \tag{C.25}
$$

We have that $\boldsymbol{d}_i \cdot \boldsymbol{r}_i = 0$; $\boldsymbol{\Lambda}_i^* \cdot \boldsymbol{c}_i = -\boldsymbol{\Lambda}_{n,i}^* \boldsymbol{r}_i \cdot \boldsymbol{c}_i$; and from the KKT,

$$\boldsymbol{r}_i \cdot (\boldsymbol{c}_i + \boldsymbol{d}_i - \boldsymbol{J}\tilde{\boldsymbol{v}}) \leq 0\,. \tag{C.26}$$

We then have that

$$\partial_{\boldsymbol{v}'} (\boldsymbol{\Lambda}_i^* \cdot \boldsymbol{d}_i) \cdot \tilde{\boldsymbol{v}} \ni \boldsymbol{\Lambda}_i^{*T} \begin{bmatrix} 0 & \hat{\boldsymbol{t}}_i^T \\ 0 & \boldsymbol{I}_2 \end{bmatrix} \boldsymbol{J}_i \tilde{\boldsymbol{v}}\,, \tag{C.27}$$

$$= \boldsymbol{\Lambda}_i^{*T} \boldsymbol{J}_i \tilde{\boldsymbol{v}} + \boldsymbol{\Lambda}_i^{*T} \begin{bmatrix} -1 & \hat{\boldsymbol{t}}_i^T \\ 0 & 0 \end{bmatrix} \boldsymbol{J}_i \tilde{\boldsymbol{v}}\,, \tag{C.28}$$

$$= \boldsymbol{\Lambda}_i^{*T} \boldsymbol{J}_i \tilde{\boldsymbol{v}} + \boldsymbol{\Lambda}_{n,i} \boldsymbol{r}_i^T \boldsymbol{J}_i \tilde{\boldsymbol{v}}\,, \tag{C.29}$$

$$= \boldsymbol{\Lambda}_i^{*T} (\boldsymbol{c}_i + \boldsymbol{d}_i) + \boldsymbol{\Lambda}_{n,i} \boldsymbol{r}_i^T \boldsymbol{J}_i \tilde{\boldsymbol{v}}\,, \tag{C.30}$$

$$= \boldsymbol{\Lambda}_i^{*T} \boldsymbol{d}_i + \boldsymbol{\Lambda}_{n,i} \boldsymbol{r}_i^T (\boldsymbol{J}_i \tilde{\boldsymbol{v}} - \boldsymbol{c}_i)\,. \tag{C.31}$$

$$= \boldsymbol{\Lambda}_i^{*T} \boldsymbol{d}_i + \boldsymbol{\Lambda}_{n,i} \boldsymbol{r}_i^T (\boldsymbol{J}_i \tilde{\boldsymbol{v}} - \boldsymbol{c}_i - \boldsymbol{d}_i)\,, \tag{C.32}$$

$$\geq \boldsymbol{\Lambda}_i^{*T} \boldsymbol{d}_i\,. \tag{C.33}$$

□

Corollary 5.

$$\boldsymbol{\Lambda}^* \cdot \boldsymbol{d} \leq \partial_{\boldsymbol{v}'} (\boldsymbol{\Lambda}^* \cdot \boldsymbol{d}) \cdot \tilde{\boldsymbol{v}}\,. \tag{C.34}$$

Proof. as each $\boldsymbol{\Lambda}_i^* \cdot \boldsymbol{d}_i$ is convex in $\boldsymbol{v}'$, $\partial_{\boldsymbol{v}'} (\boldsymbol{\Lambda}^* \cdot \boldsymbol{d}) = \partial_{\boldsymbol{v}'} \sum_i (\boldsymbol{\Lambda}_i^* \cdot \boldsymbol{d}_i) = \sum_i \partial_{\boldsymbol{v}'} (\boldsymbol{\Lambda}_i^* \cdot \boldsymbol{d}_i)$, and the claim follows from the above lemma. □

Lemma 17.

$$\boldsymbol{\Lambda}^* \cdot \boldsymbol{c} \leq \frac{\sigma_{\max}}{2} \|\tilde{\boldsymbol{v}}\|_2^2 + \sum_{i:\bar{\boldsymbol{\phi}}_i > 0} (\boldsymbol{\Lambda}_{n,i}^*)^2 \tag{C.35}$$

Proof. Consider contact i's perpendicularity constraint in the KKT:

$$0 = \boldsymbol{\Lambda}_i^* \cdot (\boldsymbol{c}_i + \boldsymbol{d}_i - \boldsymbol{J}_i \tilde{\boldsymbol{v}}) \,.$$

As $\boldsymbol{\Lambda}_i^*, \boldsymbol{d}_i \in L$ and L is self-dual, $0 \geq \boldsymbol{\Lambda}_i^* \cdot (\boldsymbol{c}_i - \boldsymbol{J}_i \tilde{\boldsymbol{v}})$. Thus,

$$\begin{aligned} \boldsymbol{\Lambda}_i^* \cdot \boldsymbol{c}_i &\leq \boldsymbol{\Lambda}_i^* \cdot \boldsymbol{J}_i \tilde{\boldsymbol{v}} \,, && \text{(C.36)} \\ &\leq \frac{1}{2} \left\| \boldsymbol{\Lambda}_i^* \right\|_2^2 + \frac{1}{2} \left\| \boldsymbol{J}_i \tilde{\boldsymbol{v}} \right\|_2^2 && \text{(C.37)} \\ &\leq (\boldsymbol{\Lambda}_{n,i}^*)^2 + \frac{1}{2} \left\| \boldsymbol{J}_i \tilde{\boldsymbol{v}} \right\|_2^2 && \text{(C.38)} \end{aligned}$$

The inequalities above come from, in order: the perpendicularity portion of the KKT; the identity $\|\boldsymbol{a}\|_2^2 + \|\boldsymbol{b}\|_2^2 \geq 2\boldsymbol{a} \cdot \boldsymbol{b}$; and the Lorentz cone constraint on $\boldsymbol{\Lambda}_i^*$, $\boldsymbol{\Lambda}_{n,i}^* \geq \left\| \boldsymbol{\Lambda}_{t,i}^* \right\|_2$.

The final inequality follows by

$$\begin{aligned} \boldsymbol{\Lambda}^* \cdot \boldsymbol{c} &= \sum_{i:\bar{\phi}_i > 0} \boldsymbol{\Lambda}_i^* \cdot \boldsymbol{c}_i \,, && \text{(C.39)} \\ &\leq \sum_{i:\bar{\phi}_i > 0} (\boldsymbol{\Lambda}_{n,i}^*)^2 + \frac{1}{2} \left\| \boldsymbol{J}_i \tilde{\boldsymbol{v}} \right\|_2^2 \,, && \text{(C.40)} \\ &= \frac{1}{2} \left\| \boldsymbol{J} \tilde{\boldsymbol{v}} \right\|_2 + \sum_{i:\bar{\phi}_i > 0} (\boldsymbol{\Lambda}_{n,i}^*)^2 \,, && \text{(C.41)} \\ &\leq \frac{\sigma_{\max}}{2} \left\| \tilde{\boldsymbol{v}} \right\|_2^2 + \sum_{i:\bar{\phi}_i > 0} (\boldsymbol{\Lambda}_{n,i}^*)^2 \,. && \text{(C.42)} \end{aligned}$$

□

We finally prove the simple inequality:

Lemma 18. *let* $|\delta| \leq \frac{1}{4}$ *and let* $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$. *Then*

$$\|\boldsymbol{b}\|_2^2 + \|\boldsymbol{a} + \delta \boldsymbol{b}\|_2^2 \geq \frac{1}{2} \|\boldsymbol{b}\|_2^2 + \frac{1}{2} \|\boldsymbol{a}\|_2^2 \,. \tag{C.43}$$

Proof. We examine two cases.

Case 1: $\|\boldsymbol{b}\|_2 \geq \|\boldsymbol{a}\|_2$. Follows trivially.

Case 2: $\|\boldsymbol{b}\|_2 < \|\boldsymbol{a}\|_2$. Follows from reverse triangle inequality as

$$\|\boldsymbol{a} + \delta\boldsymbol{b}\|_2^2 > \left\|\frac{3}{4}\boldsymbol{a}\right\|_2^2 = \frac{9}{16}\|\boldsymbol{a}\|_2^2 \tag{C.44}$$

□

C.3. Proof of Theorem 1

As $l_{\boldsymbol{\theta}}$ and therefore $\mathcal{L}_{\text{violation}}$ have minimal value zero, by definiton of quadratic growth, it suffices to show that $\mathcal{L}_{\text{violation}}$ is $2k$-QG. By Corollary 4, this reduces in the 1-datapoint case to

$$\min_{\boldsymbol{\Lambda}^* \in \mathcal{K}^*(\boldsymbol{z})} \min_{l_{\boldsymbol{\theta}}^* \in \partial_{\boldsymbol{z}} l_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{\Lambda}^*)} \|l_{\boldsymbol{\theta}}^*\|_2 \geq \sqrt{4k l_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{\Lambda}^*)}\,, \tag{C.45}$$

$$\tag{C.46}$$

Pick arbitrary $\boldsymbol{z} = [\boldsymbol{x};\ \boldsymbol{v}']$; $\boldsymbol{\Lambda}^* \in \boldsymbol{\Lambda}^*(\boldsymbol{z})$; and $l_{\boldsymbol{\theta}}^* \in \partial_{\boldsymbol{z}} l_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{\Lambda}^*)$. We can reformulate $l_{\boldsymbol{\theta}}$ in to $\tilde{l_{\boldsymbol{\theta}}}(\boldsymbol{y}(\boldsymbol{z}), \boldsymbol{\Lambda})$, where

$$\boldsymbol{y}(\boldsymbol{z}) = \begin{bmatrix} \boldsymbol{y}_{\boldsymbol{v}} \\ \boldsymbol{y}_{\boldsymbol{v}'} \\ \boldsymbol{y}_{\bar{\phi}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{v}' \\ \bar{\phi}(q, \boldsymbol{v}') \end{bmatrix}, \tag{C.47}$$

$$\tilde{l_{\boldsymbol{\theta}}}(\boldsymbol{y}, \boldsymbol{\Lambda}) = \frac{1}{2}\left\|\boldsymbol{y}_{\boldsymbol{v}'} - \boldsymbol{y}_v - \boldsymbol{\tau} - \boldsymbol{J}^T\boldsymbol{\Lambda}\right\|_2 \tag{C.48}$$

$$+ \sum_i \frac{1}{2}\text{neg}(\boldsymbol{y}_{\bar{\phi}_i})^2 + \boldsymbol{\Lambda}_i \cdot ([\text{pos}(\boldsymbol{y}_{\bar{\phi}_i}); 0] + [\|J_{t,i}\boldsymbol{y}_{\boldsymbol{v}'}\|_2\,; J_{t,i}\boldsymbol{y}_{\boldsymbol{v}'}]) \tag{C.49}$$

Then by the chain and addition rules for convex functions,

$$\hat{\partial}_{\boldsymbol{y}_v} \tilde{l_{\boldsymbol{\theta}}}(\boldsymbol{y}(\boldsymbol{z}), \boldsymbol{\Lambda}^*) = -\tilde{\boldsymbol{v}}\,, \tag{C.50}$$

$$\hat{\partial}_{\boldsymbol{y}_{\boldsymbol{v}'}} \tilde{l_{\boldsymbol{\theta}}}(\boldsymbol{y}(\boldsymbol{z}), \boldsymbol{\Lambda}^*) = \tilde{\boldsymbol{v}} + \partial_{\boldsymbol{y}_{\boldsymbol{v}'}} \left(\boldsymbol{\Lambda}^* \cdot d\right)\,, \tag{C.51}$$

$$\hat{\partial}_{\boldsymbol{y}_{\bar{\phi}}} \tilde{l_{\boldsymbol{\theta}}}(\boldsymbol{y}(\boldsymbol{z}), \boldsymbol{\Lambda}^*) = \boldsymbol{\Lambda}_n^{*,T} P(\boldsymbol{y}_{\bar{\phi}}) - \mathrm{neg}(\boldsymbol{y}_{\bar{\phi}})^T\,. \tag{C.52}$$

$\boldsymbol{y}$ is affine in its inputs with constant Jacobian

$$\nabla z = \begin{bmatrix} 0 & \boldsymbol{I} & 0 \\ 0 & 0 & \boldsymbol{I} \\ \boldsymbol{J}_n & 0 & \boldsymbol{J}_n \Delta t \end{bmatrix}\,. \tag{C.53}$$

Therefore by the subgradient chain rule Theorem (16), we may select $\boldsymbol{P}^* \in P(\boldsymbol{y}_{\bar{\phi}})$ such that

$$l_{\boldsymbol{\theta}}^* \in \left[\left(\boldsymbol{\Lambda}_n^{*,T} \boldsymbol{P}^* - \mathrm{neg}(\boldsymbol{y}_{\bar{\phi}})^T\right) \boldsymbol{J}_n \quad -\tilde{\boldsymbol{v}} \quad \tilde{\boldsymbol{v}} + \partial_{\boldsymbol{y}_{\boldsymbol{v}'}} \left(\boldsymbol{\Lambda}^* \cdot d\right) + \left(\boldsymbol{\Lambda}_n^{*,T} \boldsymbol{P}^* - \mathrm{neg}(\boldsymbol{y}_{\bar{\phi}})^T\right) \boldsymbol{J}_n \Delta t \right]\,. \tag{C.54}$$

Thus, the final inequality follows as

$$\begin{aligned}
\|l_{\boldsymbol{\theta}}^*\|_2^2 &\geq \frac{1}{2} \left\| \left(\boldsymbol{\Lambda}_n^{*,T} \boldsymbol{P}^* - \mathrm{neg}(\boldsymbol{y}_{\bar{\phi}})^T\right) \boldsymbol{J}_n \right\|_2^2 + \|\tilde{\boldsymbol{v}}\|_2^2 + \frac{1}{2} \left\| \tilde{\boldsymbol{v}} + \partial_{\boldsymbol{y}_{\boldsymbol{v}'}} \left(\boldsymbol{\Lambda}^* \cdot d\right) \right\|_2\,, && \text{(C.55)} \\
&\geq \frac{1}{2} \left\| \left(\boldsymbol{\Lambda}_n^{*,T} \boldsymbol{P}^* - \mathrm{neg}(\boldsymbol{y}_{\bar{\phi}})^T\right) \boldsymbol{J}_n \right\|_2^2 + \|\tilde{\boldsymbol{v}}\|_2^2 + \partial_{\boldsymbol{y}_{\boldsymbol{v}'}} \left(\boldsymbol{\Lambda}^* \cdot d\right) \cdot \tilde{\boldsymbol{v}}\,, && \text{(C.56)} \\
&\geq \frac{1}{2} \left\| \left(\boldsymbol{\Lambda}_n^{*,T} \boldsymbol{P}^* - \mathrm{neg}(\boldsymbol{y}_{\bar{\phi}})^T\right) \boldsymbol{J}_n \right\|_2^2 + \|\tilde{\boldsymbol{v}}\|_2^2 + \boldsymbol{\Lambda}^* \cdot d\,, && \text{(C.57)} \\
&\geq \frac{\sigma_{\min}}{2} \left\| \boldsymbol{\Lambda}_n^{*,T} \boldsymbol{P}^* - \mathrm{neg}(\boldsymbol{y}_{\bar{\phi}}) \right\|_2^2 + \|\tilde{\boldsymbol{v}}\|_2^2 + \boldsymbol{\Lambda}^* \cdot d\,, && \text{(C.58)} \\
&\geq \frac{\sigma_{\min}}{2} \left(\left\| \boldsymbol{\Lambda}_n^{*,T} \boldsymbol{P}^* \right\|_2 + \left\| \mathrm{neg}(\boldsymbol{y}_{\bar{\phi}}) \right\|_2^2 \right) + \|\tilde{\boldsymbol{v}}\|_2^2 + \boldsymbol{\Lambda}^* \cdot d\,, && \text{(C.59)} \\
&\geq \frac{\sigma_{\min}}{2} \left(\sum_{i:\bar{\phi}_i > 0} (\boldsymbol{\Lambda}_{n,i}^*)^2 + \left\| \mathrm{neg}(\boldsymbol{y}_{\bar{\phi}}) \right\|_2^2 \right) + \|\tilde{\boldsymbol{v}}\|_2^2 + \boldsymbol{\Lambda}^* \cdot d\,, && \text{(C.60)} \\
&\geq \frac{\sigma_{\min}}{2 \max(1, \sigma_{\max})} l_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{\Lambda})\,. && \text{(C.61)}
\end{aligned}$$

These inequalities follow in order by: Lemma 18; the identity $\|\boldsymbol{a} + \boldsymbol{b}\|_2^2 \geq 2\boldsymbol{a} \cdot \boldsymbol{b}$; Corollary 5; $\boldsymbol{J}_n \boldsymbol{J}_n^T \succeq \sigma_{\min} \boldsymbol{I}$; the identity $P(\bar{\boldsymbol{\phi}}) \cdot \mathrm{neg}(\bar{\boldsymbol{\phi}}) = 0$; the definition of $\partial \mathrm{pos} = P$; and the identity

$\sigma_{\max} \boldsymbol{I} \succeq \boldsymbol{J}^T \boldsymbol{J}$.

APPENDIX D

PROOFS FOR CHAPTER 6

D.1. Proof of Theorem 2

From (6.1) we have that $s(\hat{\boldsymbol{r}}) = \max_i \boldsymbol{p}_i \cdot \hat{\boldsymbol{r}}$. Let $\boldsymbol{P}_k \in \mathbb{R}^{M\times 3}$ be the matrix with ith row equal to $\boldsymbol{p}_{i+k(\mathrm{mod}M)}$. Therefore,

$$\max(\boldsymbol{P}_0\hat{\boldsymbol{r}}, \ldots, \boldsymbol{P}_{M-1}\hat{\boldsymbol{r}}) = s(\hat{\boldsymbol{r}})\mathbf{1}_M\,, \tag{D.1}$$

where $\max(\cdot)$ is an element-wise max between each of its inputs. We will make frequent use of the identity $\max(\boldsymbol{a}, \boldsymbol{b}) = \mathrm{pos}(\boldsymbol{a} - \boldsymbol{b}) + \boldsymbol{b}$.

Set the parameters of the H-FICNN as follows:

$$\boldsymbol{W}_{\boldsymbol{r},k} = \boldsymbol{P}_{k-1} - \boldsymbol{P}_k\,, k \in 1, \ldots, M-1\,, \tag{D.2}$$

$$\boldsymbol{W}_{\boldsymbol{r},M} = \frac{1}{M}\mathbf{1}_M^T\boldsymbol{P}_{M-1}\,, \tag{D.3}$$

$$\boldsymbol{W}_{\boldsymbol{z},k} = \boldsymbol{I}_M\,, k \in 2, \ldots, M-1\,, \tag{D.4}$$

$$\boldsymbol{W}_{\boldsymbol{z},M} = \frac{1}{M}\mathbf{1}_M^T\,, \tag{D.5}$$

Plugging in to the network equations (2.51), we now show by induction that each intermediate layer has value $\boldsymbol{z}_k = \max(\boldsymbol{P}_0\hat{\boldsymbol{r}}, \ldots, \boldsymbol{P}_k\hat{\boldsymbol{r}}) - \boldsymbol{P}_k\hat{\boldsymbol{r}}$ for $k \in 1, \ldots, M-1$.

Base case ($k = 1$):

$$\boldsymbol{z}_1 = \boldsymbol{\sigma}(\boldsymbol{W}_{\boldsymbol{r},1}\hat{\boldsymbol{r}})\,, \tag{D.6}$$

$$= \mathrm{pos}(\boldsymbol{P}_0\hat{\boldsymbol{r}} - \boldsymbol{P}_1\hat{\boldsymbol{r}})\,, \tag{D.7}$$

$$= \max(\boldsymbol{P}_0\hat{\boldsymbol{r}}, \boldsymbol{P}_1\hat{\boldsymbol{r}}) - \boldsymbol{P}_1\hat{\boldsymbol{r}}\,. \tag{D.8}$$

$$\tag{D.9}$$

Inductive case ($k-1 \implies k$):

$$\begin{aligned}
\boldsymbol{z}_k &= \boldsymbol{\sigma}(\boldsymbol{W}_{\boldsymbol{z},k}\boldsymbol{z}_{k-1} + \boldsymbol{W}_{\boldsymbol{r},k}\hat{\boldsymbol{r}})\,, && \text{(D.10)}\\
&= \mathrm{pos}(\boldsymbol{z}_{k-1} + (\boldsymbol{P}_{k-1} - \boldsymbol{P}_k)\hat{\boldsymbol{r}})\,, && \text{(D.11)}\\
&= \mathrm{pos}(\max(\boldsymbol{P}_0\hat{\boldsymbol{r}}, \ldots, \boldsymbol{P}_{k-1}\hat{\boldsymbol{r}}) - \boldsymbol{P}_{k-1}\hat{\boldsymbol{r}} + (\boldsymbol{P}_{k-1} - \boldsymbol{P}_k)\hat{\boldsymbol{r}})\,, && \text{(D.12)}\\
&= \mathrm{pos}(\max(\boldsymbol{P}_0\hat{\boldsymbol{r}}, \ldots, \boldsymbol{P}_{k-1}\hat{\boldsymbol{r}}) - \boldsymbol{P}_k\hat{\boldsymbol{r}})\,, && \text{(D.13)}\\
&= \max(\boldsymbol{P}_0\hat{\boldsymbol{r}}, \ldots, \boldsymbol{P}_k\hat{\boldsymbol{r}}) - \boldsymbol{P}_k\hat{\boldsymbol{r}}\,. && \text{(D.14)}
\end{aligned}$$

Therefore,

$$\begin{aligned}
s_{\boldsymbol{\theta}}(\hat{\boldsymbol{r}}) &= \boldsymbol{W}_{\boldsymbol{r},M}\boldsymbol{z}_{M-1} + \boldsymbol{W}_{\boldsymbol{r},M}\hat{\boldsymbol{r}}\,, && \text{(D.15)}\\
&= \frac{1}{M}\mathbf{1}_M^T\left(\max(\boldsymbol{P}_0\hat{\boldsymbol{r}}, \ldots, \boldsymbol{P}_{M-1}\hat{\boldsymbol{r}}) - \boldsymbol{P}_{M-1}\hat{\boldsymbol{r}} + P_{M-1}\hat{\boldsymbol{r}}\right)\,, && \text{(D.16)}\\
&= \frac{1}{M}\mathbf{1}_M^T\left(\max(\boldsymbol{P}_0\hat{\boldsymbol{r}}, \ldots, \boldsymbol{P}_{M-1}\hat{\boldsymbol{r}})\right)\,, && \text{(D.17)}\\
&= \frac{\mathbf{1}_M^T\mathbf{1}_M}{M}s(\hat{\boldsymbol{r}})\,, && \text{(D.18)}\\
&= s(\hat{\boldsymbol{r}})\,. && \text{(D.19)}
\end{aligned}$$

D.2. Proof of Theorem 3

It is sufficient to rasterize the shape. Let $N = \left\lceil \frac{2D\sqrt{3}}{\varepsilon} \right\rceil^3$, and decompose $[-D, D]^3$ into a $N^{\frac{1}{3}} \times N^{\frac{1}{3}} \times N^{\frac{1}{3}}$ grid of cube voxels $\mathcal{S}_1, \ldots, \mathcal{S}_N$.

Each voxel has side-length $\leq \frac{\varepsilon}{\sqrt{3}}$, and thus no two points inside any voxel are more than ε apart. Consider the union of (convex polytope) voxel cubes

$$\bar{\mathcal{S}} = \bigcup_{i:\mathcal{S}_i \cap \mathcal{S} \neq \emptyset} \mathcal{S}_i\,. \tag{D.20}$$

It is clear to see that $\bar{\mathcal{S}} \supseteq \mathcal{S}$, so

$$\max_{s \in \mathcal{S}} \mathrm{Dist}(s, \bar{\mathcal{S}}) = 0\,. \tag{D.21}$$

Any $\bar{s} \in \bar{\mathcal{S}}$ is by definition in a voxel $\mathcal{S}_i$ containing some point $s(\bar{s}) \in \mathcal{S} \cap \mathcal{S}_i$, and thus

$$\max_{\bar{s} \in \bar{\mathcal{S}}} \mathrm{Dist}(\bar{s}, \mathcal{S}) \leq \max_{\bar{s} \in \bar{\mathcal{S}}} \mathrm{Dist}(\bar{s}, s(\bar{s})) \leq \varepsilon \,. \tag{D.22}$$

APPENDIX E

EXPERIMENT DETAILS AND PROOFS FOR CHAPTER 7

E.1. Example Details

Here, we list relevant details on the examples in Section 7.5. The MATLAB codebase at `https://github.com/mshalm/routh-multi-impact` may be run via `Results()`. The PATH LCP solver (Dirkse and Ferris, 1995) must be installed, and `pathlcp.m` must be available from the MATLAB path. Geometric, inertial, and simulation parameters of the examples are listed in Tables E.1–E.5, and the listed symbols match the variable names used in the codebase. Unless otherwise stated, bodies have uniform density, and links are massless. For the RAMone example, we refer the reader to Remy (2017) for a full description of the system's inertial and geometric properties.

Table E.1: Block Drop Parameters

Parameter	Symbol	Value
Phone width	a	$7.444\,\mathrm{cm}$
Phone height	b	$16.094\,\mathrm{cm}$
Phone mass	m	$190\,\mathrm{g}$
Init. downward vel.	v_0	$14.01\,\mathrm{cm\,s^{-1}}$
Friction coefficient	μ	1
Step size	h	$0.3\,\mathrm{N\,s}$
Trajectory length	N	10
Trajectory set size	M	2^{14}

Table E.2: Compass-gait Walker Parameters

Parameter	Symbol	Value
Leg length	l	$1\,\mathrm{m}$
Mass-to-foot length	$s_{\parallel}$	$0.5\,\mathrm{m}$
Leg mass	m	$1\,\mathrm{kg}$
Trailing leg pitch	φ_{tr}	$78°$
Leading leg pitch	φ_{le}	$-78°$
Trailing leg init. ang. vel.	$\dot{\varphi}_{tr,0}$	$0.5\,\mathrm{rad\,s^{-1}}$
Leading leg init. ang. vel.	$\dot{\varphi}_{le,0}$	$0.5\,\mathrm{rad\,s^{-1}}$
Friction coefficient	μ	5
Step size	h	$1\,\mathrm{N\,s}$
Trajectory length	N	5
Trajectory set size	M	2^{20}

Table E.3: Box-Wall Parameters

Parameter	Symbol	Value
Box side length	w	$1\,\mathrm{m}$
Box mass	m	$1\,\mathrm{kg}$
Angle from box to ground	θ	$10°$
Init. horizontal velocity	v_0	$1\,\mathrm{m\,s^{-1}}$
Friction coefficient	μ	1
Step size	h	$2\,\mathrm{N\,s}$
Trajectory length	N	5
Trajectory set size	M	2^{18}

Table E.4: RAMone Parameters

Parameter	Symbol	Value
Trunk pitch	Φ	$16°$
Leading hip angle	α_{le}	$-70°$
Trailing hip angle	α_{tr}	$70°$
Leading knee angle	β_{le}	$-2°$
Trailing knee angle	β_{tr}	$-92.48°$
Trunk init. x vel.	$\dot{x}_0$	$-0.4114\,\mathrm{m\,s^{-1}}$
Trunk init. y vel.	$\dot{y}_0$	$-0.2105\,\mathrm{m\,s^{-1}}$
Trunk init. ang. vel.	$\dot{\Phi}_0$	$1\,\mathrm{rad\,s^{-1}}$
Leading hip init. vel.	$\dot{\alpha}_{le,0}$	$0\,\mathrm{rad\,s^{-1}}$
Trailing hip init. vel.	$\dot{\alpha}_{tr,0}$	$0\,\mathrm{rad\,s^{-1}}$
Leading knee init. vel.	$\dot{\beta}_{le,0}$	$0\,\mathrm{rad\,s^{-1}}$
Trailing knee init. vel.	$\dot{\beta}_{tr,0}$	$0\,\mathrm{rad\,s^{-1}}$
Friction coefficient	μ	10^5
Step size	h	$1\,\mathrm{N\,s}$
Trajectory length	N	10
Trajectory set size	M	2^{20}

Table E.5: Disk Stacking Parameters

Parameter	Symbol	Value
Disk radius	R	$1\,\mathrm{m}$
Disk mass	m	$1\,\mathrm{kg}$
Initial vertical velocity	v_0	$-1\,\mathrm{m\,s^{-1}}$
Friction coefficient	μ	$\sqrt{3}$
Step size	h	$1\,\mathrm{N\,s}$
Trajectory length	N	10
Trajectory set size	M	2^{20}

E.2. Background Proofs

E.2.1. Proof of Proposition 5

Let $g(x) : \mathbb{R}^n \to \mathbb{R}^m$ be Lipschitz with constant L and let $h > 0$. Let $\mathcal{X} = \{x_1, \ldots, x_N\}$ be a set of uniform i.i.d. samples from $[0, h]^n$. By Lipschitz continuity, $g(\mathcal{X})$ is a ε-net of $g([0, h]^n)$ if $\mathcal{X}$ is an $\frac{\varepsilon}{L}$-net of $[0, h]^n$; we examine the latter.

Consider regularly-spaced grid points of cardinality M^n:

$$\mathcal{X}' = \left\{ \tfrac{h}{2M}, \tfrac{3h}{2M}, \ldots, \tfrac{(2M-1)h}{2M} \right\}^n . \tag{E.1}$$

$\mathcal{X}'$ is a $\frac{h\sqrt{n}}{2M}$-net of $[0, h]^n$. Thus, setting $M = \left\lceil \frac{hL\sqrt{n}}{\varepsilon} \right\rceil$, $\mathcal{X}'$ is an $\frac{\varepsilon}{2L}$-net of $[0, h]^n$. Consider the case where for each $x \in \mathcal{X}'$, $\mathcal{X}$ contains an x_i with

$$x_i \in x + \left[-\tfrac{h}{2M}, \tfrac{h}{2M} \right]^n \subseteq [0, h]^n , \tag{E.2}$$

and thus $\|x_i - x\|_2 \leq \frac{\varepsilon}{2L}$. Then by triangle inequality, $\mathcal{X}$ is an $\frac{\varepsilon}{L}$-net of $[0, h]^n$ when (E.2) holds for each x_i. For a single $x \in \mathcal{X}'$, as the elements of $\mathcal{X}$ are chosen uniform i.i.d, the probability of (E.2) *not* holding is $(1 - M^{-n})^N$. The probability of (E.2) holding for every x is at least

$$1 - M^n (1 - M^{-n})^N , \tag{E.3}$$

by union bound. The proof holds as $M^{-n} = \Omega$.

E.2.2. Proof of Lemma 1

We may assume WLOG that $\boldsymbol{M} = \boldsymbol{I}$ by applying a coordinate transformation of $\boldsymbol{M}^{\frac{1}{2}}$ to $\boldsymbol{v}$. Let $\boldsymbol{R}$ be a matrix with columns that constitute an orthogonal basis of Range $\left(\boldsymbol{J}_i^T\right)$. By equivalence of norms there exists $\varepsilon > 0$ such that

$$\|\boldsymbol{J}_{n,i}\boldsymbol{v}\| + \|\boldsymbol{J}_{t,i}\boldsymbol{v}\|_2 \geq \varepsilon \left\|\boldsymbol{R}^T\boldsymbol{v}\right\|_2 . \tag{E.4}$$

We will show that $\kappa = (\varepsilon \min(\boldsymbol{\mu}_i, 1))^{-1}$ satisfies the claim. Let $V(s) = \left\|\boldsymbol{R}^T \boldsymbol{v}(s)\right\|_2^2$. Assume WLOG that $\boldsymbol{v}(s)$ is a colling velocity ($\boldsymbol{v}(s) \in \mathcal{C}(\boldsymbol{q})$) at least until $s^* = \left\|\boldsymbol{R}^T \boldsymbol{v}(0)\right\|_2 \kappa \leq \|\boldsymbol{v}(0)\|_2 \kappa$. Then, on the interval $[0, s^*)$,

$$\dot{V} = 2\dot{\boldsymbol{v}}^T \boldsymbol{R}\boldsymbol{R}^T \boldsymbol{v}\,, \tag{E.5}$$

$$\in 2\left(\boldsymbol{J}_{n,i} - \boldsymbol{\mu}_i \text{Unit}\left(\boldsymbol{J}_{t,i}\boldsymbol{v}\right)^T \boldsymbol{J}_{t,i}\right) \boldsymbol{R}\boldsymbol{R}^T \boldsymbol{v}\,, \tag{E.6}$$

$$= -2\left\|\boldsymbol{J}_{n,i}\boldsymbol{v}\right\| - 2\boldsymbol{\mu}_i \left\|\boldsymbol{J}_{t,i}\boldsymbol{v}\right\|_2\,, \tag{E.7}$$

$$\leq -2\varepsilon \min(\boldsymbol{\mu}_i, 1)\sqrt{V}\,, \tag{E.8}$$

$$\leq -\frac{2}{\kappa}\sqrt{V}\,. \tag{E.9}$$

The unique solution to the IVP $\dot{x} = -\frac{2}{\kappa}\sqrt{x}$,

$$x(s) = \left(\sqrt{x(0)} - \frac{s}{\kappa}\right)^2\,, \tag{E.10}$$

therefore bounds V from above on $[0, s^*)$. Thus,

$$V(s^*) \leq \left(\sqrt{V(0)} - \frac{s^*}{\kappa}\right)^2\,, \tag{E.11}$$

$$= \left(\left\|\boldsymbol{R}^T \boldsymbol{v}(0)\right\|_2 - \left\|\boldsymbol{R}^T \boldsymbol{v}(0)\right\|_2\right)^2\,, \tag{E.12}$$

$$= 0\,. \tag{E.13}$$

Therefore $\boldsymbol{R}^T \boldsymbol{v}(s^*) = \boldsymbol{0}$, $\boldsymbol{J}_{n,i}\boldsymbol{v}(s^*) = 0$, and $\boldsymbol{v}(s^*) \notin \mathcal{C}(\boldsymbol{q})$.

E.3. Impact Model Proofs

E.3.1. Proof of Lemma 3

The final claim may be reached via direct application of Theorem 15, as long as $D_{\boldsymbol{q}}(\boldsymbol{v})$ is non-empty, uniformly bounded, closed-valued, convex-valued, and u.s.c. We will demonstrate that each of these properties hold.

We first observe that the set of contacts $\mathcal{I}_{\boldsymbol{q}}(\boldsymbol{v})$, used in the construction of $D_{\boldsymbol{q}}(\boldsymbol{v})$ in (7.3), is

non-empty by construction. Furthermore, $\mathcal{I}_{\boldsymbol{q}}(\boldsymbol{v})$ is u.s.c. in $\boldsymbol{v}$, because it is constructed from non-strict inequalities of linear functions of $\boldsymbol{v}$. Next, we note that for each i, $F_i(\boldsymbol{q}, \boldsymbol{v}, 1)$ is non-empty, uniformly bounded, closed-valued, and u.s.c. as it is an affine transformation of Unit$(\cdot)$. Finally, we characterize $D_{\boldsymbol{q}}(\boldsymbol{v})$. $D_{\boldsymbol{q}}(\boldsymbol{v})$ is non-empty, uniformly bounded, and close-convex valued, by construction from the convex hull of a non-empty union of $F_i(\boldsymbol{q}, \boldsymbol{v}, 1)$. Now, consider an arbitrary velocity $\boldsymbol{v}_0$ and neighborhood $\dot{\mathcal{V}}_0 \supset D_{\boldsymbol{q}}(\boldsymbol{v}_0)$. As $\mathcal{I}_{\boldsymbol{q}}(\boldsymbol{v})$ is u.s.c., we can select a neighborhood $\mathcal{V}$ with $\mathcal{I}_{\boldsymbol{q}}(\mathcal{V}) \subseteq \mathcal{I}_{\boldsymbol{q}}(\boldsymbol{v}_0)$. Therefore on $\mathcal{V}$,

$$D_{\boldsymbol{q}}(\boldsymbol{v}) \subseteq D_0(\boldsymbol{v}) = \text{co}\left(\cup_{i \in \mathcal{I}_{\boldsymbol{q}}(\boldsymbol{v}_0)} F_i(\boldsymbol{q}, \boldsymbol{v}, 1)\right) . \tag{E.14}$$

$D_0(\boldsymbol{v})$ is u.s.c. as the convex hull of u.s.c. functions, and furthermore $D_{\boldsymbol{q}}(\boldsymbol{v}_0) = D_0(\boldsymbol{v}_0)$. Therefore by definition of u.s.c. there exists a neighborhood $\mathcal{V}_0$ of $\boldsymbol{v}_0$ such that

$$D_{\boldsymbol{q}}(\mathcal{V}_0) \subseteq D_0(\mathcal{V}_0) \subseteq \dot{\mathcal{V}}_0 . \tag{E.15}$$

$D_{\boldsymbol{q}}(\mathcal{V})$ is therefore by definition u.s.c. and the claim is satified.

E.3.2. Proof of Lemma 4

Consider a configuration $\boldsymbol{q} \in \mathbb{R}^{n_q}$ and compact interval $[a, b]$. We first demonstrate that the impact DI mapping $D_{\boldsymbol{q}}(\boldsymbol{v})$ is positively homogeneous in $\boldsymbol{v}$ ($D_{\boldsymbol{q}}(\boldsymbol{v}) = D_{\boldsymbol{q}}(k\boldsymbol{v})$ for $k > 0$). The DI mapping $F_i(\boldsymbol{q}, \boldsymbol{v}, 1)$ for contact i is an affine transform of Unit$(\boldsymbol{J}_{t,i}\boldsymbol{v})$ and thus positively homogeneous. Also, the the set of contacts $\mathcal{I}_{\boldsymbol{q}}(\boldsymbol{v})$ used in the construction of $D_{\boldsymbol{q}}(\boldsymbol{v})$ in (7.3), is also positively homogeneous in $\boldsymbol{v}$. Therefore, $D_{\boldsymbol{q}}(\boldsymbol{v}) = \boldsymbol{M}^{-1}\text{co}\left(\cup_{i \in \mathcal{I}_{\boldsymbol{q}}} F_i\right)$ is positively homogeneous.

Now, consider a solution $\boldsymbol{v}(s)$ to the impact DI $\dot{\boldsymbol{v}} \in D_{\boldsymbol{q}}(\boldsymbol{v})$ over $[a, b]$, and $k > 0$. $k\boldsymbol{v}(\frac{s}{k})$ is well-defined and absolutely continuous over the interval $[ka, kb]$, and has derivative equal to $\dot{\boldsymbol{v}}(\frac{s}{k})$ a.e. on $[ka, kb]$. Then $\dot{\boldsymbol{v}}(\frac{s}{k}) \in D_{\boldsymbol{q}}(\boldsymbol{v}(\frac{s}{k})) = D_{\boldsymbol{q}}(k\boldsymbol{v}(\frac{s}{k}))$ a.e., and $k\boldsymbol{v}(\frac{s}{k}) \in \text{SOL}\left(D_{\boldsymbol{q}}, [ka, kb]\right)$.

E.3.3. Proof of Lemma 5

Let $\boldsymbol{q} \in \mathcal{Q}_A$, and let $[a, b]$ be a compact interval. Consider a solution of the impact DI $\boldsymbol{v}(s) \in \text{SOL}\left(D_{\boldsymbol{q}}, [a, b]\right)$ with non-separating velocity ($\boldsymbol{v}([a, b]) \subseteq \text{cl}\mathcal{C}(\boldsymbol{q})$). We will show that $\|\boldsymbol{v}(s)\|_{\boldsymbol{M}}$ is

non-increasing by proving that $\dot{K}(\boldsymbol{q}, \boldsymbol{v}(s))$ is non-positive almost everywhere. Pick any $s \in [a, b]$ where $\dot{\boldsymbol{v}}(s) \in D_{\boldsymbol{q}}(\boldsymbol{v}(s))$. By construction of $D_{\boldsymbol{q}}(\boldsymbol{v})$ (7.3) and the definition of the convex hull, there exists coefficients $c_i \geq 0$ such that

$$\boldsymbol{M}\dot{\boldsymbol{v}}(s) \in \sum_{i:\boldsymbol{J}_{n,i}\boldsymbol{v}(s)\leq 0} c_i F_i(\boldsymbol{q}, \boldsymbol{v}(s), 1)\,. \tag{E.16}$$

We observe by chain rule that

$$\dot{K} = \boldsymbol{v}^T \boldsymbol{M}\dot{\boldsymbol{v}} \in \sum_{i:\boldsymbol{J}_{n,i}\boldsymbol{v}\leq 0} c_i \boldsymbol{v}^T F_i(\boldsymbol{q}, \boldsymbol{v}, 1)\,. \tag{E.17}$$

$\dot{K}$ is then non-positive as each term in this sum is non-positive by construction of F_i and (7.2):

$$\boldsymbol{v}^T F_i(\boldsymbol{q}, \boldsymbol{v}, 1) = \boldsymbol{v}^T \boldsymbol{J}_{n,i}^T - \boldsymbol{\mu}_i \left\|\boldsymbol{J}_{t,i}\boldsymbol{v}\right\|_2\,. \tag{E.18}$$

E.3.4. Proof of Theorem 4

Let $\boldsymbol{q} \in \mathcal{Q}_A$ be a configuration with active contact, and $\boldsymbol{v}(s) \in \mathrm{SOL}\left(D_{\boldsymbol{q}}, [a, b]\right)$ a solution to the associated impact DI with impacting velocity ($\boldsymbol{v}([a, b]) \subseteq \mathrm{cl}\mathcal{C}(\boldsymbol{q})$). Let $\boldsymbol{\lambda}(s)$ be the associated vector of force variables defined a.e..

Assume that $\boldsymbol{v}(s)$ is non-constant. We may now prove the claim by showing that $\|\boldsymbol{v}(b)\|_{\boldsymbol{M}} < \|\boldsymbol{v}(a)\|_{\boldsymbol{M}}$. As $\boldsymbol{v}(s)$ is continuous, we may select $a < s^* < b$ such that $\forall \delta > 0$, $\boldsymbol{v}(s)$ is not constant on $[s^*, s^* + \delta]$. Let $A = \{i \in \mathcal{I}_A(\boldsymbol{q}) : \boldsymbol{J}_{n,i}\boldsymbol{v}(s^*) \leq 0\}$ be the set of non-separating contacts at $s = s^*$. Let B be the set of contacts $b \in A$ with zero contact velocity ($\boldsymbol{J}_b\boldsymbol{v}(s^*) = \boldsymbol{0}$). As $\boldsymbol{v}(s)$ is continuous, $\exists \delta_\varepsilon > 0$ and $\varepsilon > 0$ such that $\forall s \in [s^*, s^* + \delta_\varepsilon] \subseteq [a, b]$,

- All $i \in \mathcal{I}_A \setminus A$ separate ($\boldsymbol{J}_{n,i}\boldsymbol{v}(s) > \varepsilon$)
- All $i \in A \setminus B$ move: $\boldsymbol{J}_{n,i}\boldsymbol{v}(s) < -\varepsilon$ or $\left\|\boldsymbol{J}_{t,i}\boldsymbol{v}(s)\right\|_2 > \frac{\varepsilon}{\boldsymbol{\mu}_i}$.

Select s from $[s^*, s^* + \delta_\varepsilon]$ with $\boldsymbol{v}(s) \neq \boldsymbol{v}(s^*)$. By Lemma 5,

$$0 \geq \|\boldsymbol{v}(s)\|_{\boldsymbol{M}}^2 - \|\boldsymbol{v}(s^*)\|_{\boldsymbol{M}}^2 , \tag{E.19}$$

$$= 2\boldsymbol{v}(s^*)^T \boldsymbol{M} \left(\boldsymbol{v}(s) - \boldsymbol{v}(s^*)\right) + \|\boldsymbol{v}(s) - \boldsymbol{v}(s^*)\|_{\boldsymbol{M}}^2 , \tag{E.20}$$

$$= 2 \left(\boldsymbol{J}\boldsymbol{v}(s^*)\right)^T \int_{s^*}^{s} \lambda(\sigma) \mathrm{d}\sigma + \|\boldsymbol{v}(s) - \boldsymbol{v}(s^*)\|_{\boldsymbol{M}}^2 . \tag{E.21}$$

Therefore, there must exist a contact $a \in A \setminus B$ with $\int_{s^*}^{s} \boldsymbol{\lambda}_{n,a} > 0$ as (E.21) is non-positive. Finally,

$$K(\boldsymbol{v}(s)) = K(\boldsymbol{v}(s^*)) + \int_{s^*}^{s} (\boldsymbol{J}\boldsymbol{v}(\tau))^T \boldsymbol{\lambda}(\sigma) \mathrm{d}\sigma , \tag{E.22}$$

$$\leq K(\boldsymbol{v}(s^*)) - \varepsilon \int_{s^*}^{s} \boldsymbol{\lambda}_{n,a}(\sigma) \mathrm{d}\sigma , \tag{E.23}$$

$$< K(\boldsymbol{v}(s^*)) . \tag{E.24}$$

Therefore, as K is non-increasing, $\|\boldsymbol{v}(b)\|_{\boldsymbol{M}} < \|\boldsymbol{v}(a)\|_{\boldsymbol{M}}$.

E.3.5. Proof of Lemma 6

Suppose not, so there exists a configuration $\boldsymbol{q} \in \mathcal{Q}_A$, dissipation rate $\alpha_{\boldsymbol{q}}(s)$ such that $\dot{\boldsymbol{v}} \in D_{\boldsymbol{q}}(\boldsymbol{v})$ is $\alpha_{\boldsymbol{q}}(s)$-dissipative, $s > 0$ and $s^* > \|\boldsymbol{v}(0)\|_{\boldsymbol{M}} \frac{s}{\alpha_{\boldsymbol{q}}(s)}$, and $\boldsymbol{v}(s) \in \mathrm{SOL}\left(D_{\boldsymbol{q}}, [0, s^*]\right)$ with $\boldsymbol{v}([0, s^*]) \subseteq \mathrm{cl}\mathcal{C}(\boldsymbol{q})$. Assume WLOG by Lemma 4 that $\|\boldsymbol{v}(0)\|_{\boldsymbol{M}} = 1$. By solution homogeneity (Lemma 4) we have for any $s_k = s_{k-1} + s \|v(s_k)\|_{\boldsymbol{M}}$,

$$\frac{\|\boldsymbol{v}\left(s_k\right)\|_{\boldsymbol{M}}}{\|\boldsymbol{v}\left(s_{k-1}\right)\|_{\boldsymbol{M}}} \leq \left(1 - \alpha_{\boldsymbol{q}}(s)\right) . \tag{E.25}$$

Setting $s_0 = 0$, we thus have

$$\|\boldsymbol{v}\left(s_k\right)\|_{\boldsymbol{M}} \leq \left(1 - \alpha_{\boldsymbol{q}}(s)\right)^k , \tag{E.26}$$

$$s_k \leq s \sum_{i=1}^{k} \left(1 - \alpha_{\boldsymbol{q}}(s)\right)^{k-1} . \tag{E.27}$$

Therefore $s_\infty = \lim_{k \to \infty} s_k = \frac{s}{\alpha_{\boldsymbol{q}}(s)} < s^*$ and by continuity of $\boldsymbol{v}$, $\boldsymbol{v}\left(s_\infty\right) = \boldsymbol{0}$. But then by Theorem 5 $\|\boldsymbol{v}(s)\|_{\boldsymbol{M}}$ must decrease below 0 on $[s_\infty, s^*]$, a contradiction.

E.3.6. Proof of Theorem 6

Suppose not. Then there exists a $\boldsymbol{q} \in \mathcal{Q}_A \setminus \mathcal{Q}_P$, an $s_f > 0$, and a corresponding sequence of solutions $\left(\boldsymbol{v}^j(s)\right)_{j\in\mathbb{N}}$, $\boldsymbol{v}^j(s) \in \mathrm{SOL}\left(D_{\boldsymbol{q}}, [0, s_f]\right)$, all starting with $\left\|\boldsymbol{v}^j(0)\right\|_{\boldsymbol{M}} = 1$ and never exiting $\mathrm{cl}\mathcal{C}(\boldsymbol{q})$, which dissipate less and less energy:

$$\lim_{j\to\infty} \left\|\boldsymbol{v}^j(s_f)\right\|_{\boldsymbol{M}} = 1\,. \tag{E.28}$$

As $D_{\boldsymbol{q}}$ is bounded and each solution $\boldsymbol{v}^j(s)$ never exits $\left\{\left\|\boldsymbol{v}^j(s)\right\|_{\boldsymbol{M}} \leq 1\right\}$ by dissipation (Lemma 5), the sequence is equicontinuous and bounded. By Theorem 15, a subsequence of $\boldsymbol{v}^j(s)$ converges uniformly to a function $\boldsymbol{v}_\infty(s)$, with $\|\boldsymbol{v}_\infty(s_f)\|_{\boldsymbol{M}} = 1$ by (E.28). Because kinetic energy is non-increasing, $\|\boldsymbol{v}_\infty(s)\|_{\boldsymbol{M}} = 1$ for all $s \in [0, s_f]$. By Lemma 3 $\boldsymbol{v}_\infty(s)$ is a solution to $\dot{\boldsymbol{v}} \in D_{\boldsymbol{q}}(\boldsymbol{v})$. Therefore as $\boldsymbol{v}_\infty(s)$ does not dissipate kinetic energy, it is constant by Theorem 4, and thus $\mathbf{0} \in D_{\boldsymbol{q}}(\boldsymbol{v}_\infty(s))$. But as each $\boldsymbol{v}^j(s) \in \mathrm{cl}\mathcal{C}(\boldsymbol{q})$, we must also have $\boldsymbol{v}_\infty(s) \in \mathrm{cl}\mathcal{C}(\boldsymbol{q})$, which contradicts Assumption 2.

E.3.7. Proof of Corollary 1

Let $\mathcal{Q} \subseteq \mathcal{Q}_A \setminus \mathcal{Q}_P$ be a compact set of non-penetrating configurations with active contact. We will construct a suitable $\alpha_{\mathcal{Q}}$ explicitly. Let $s_f > 0$. Define the DI

$$\begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \dot{\boldsymbol{x}} \in D'(\boldsymbol{x}) = \begin{bmatrix} \mathbf{0} \\ D_{\boldsymbol{q}}(\boldsymbol{v}) \end{bmatrix}. \tag{E.29}$$

As the set of active contacts $\mathcal{I}_A(\boldsymbol{q})$ is u.s.c. and $\boldsymbol{M}, \boldsymbol{J}$ are continuous, D' is compact-convex, uniformly bounded, and u.s.c.. Now consider the sets

$$\mathcal{X}_0 = \left\{ \begin{bmatrix} \boldsymbol{q}_0 \\ \boldsymbol{v}_0 \end{bmatrix} : \boldsymbol{q}_0 \in \mathcal{Q} \wedge \|\boldsymbol{v}_0\|_{\boldsymbol{M}(\boldsymbol{q}_0)} = 1 \right\}, \tag{E.30}$$

$$\mathcal{X}_f = \left\{\boldsymbol{x}(s_f) : \boldsymbol{x}(s) \in \mathrm{IVP}\left(D', \mathcal{X}_0, [0, s_f]\right)\right\}. \tag{E.31}$$

$\mathcal{X}_0$ represents all initial conditions with configurations in $\mathcal{Q}$ and initial kinetic energy $\frac{1}{2}$, and $\mathcal{X}_f$ is set of states reachable from $\mathcal{X}_0$ via solutions to the dynamics (E.29) for a duration s_f. As $\mathcal{X}_0$ is compact, IVP $(D', \mathcal{X}_0, [0, s_f])$ and therefore $\mathcal{X}_f$ is closed and non-empty by Proposition 2. Any solution $[\boldsymbol{q}(s);\ \boldsymbol{v}(s)] \in \text{IVP}\,(D', \mathcal{X}_0, [0, s_f])$ must have constant $\boldsymbol{q}(s) = \boldsymbol{q}(0) \in \mathcal{Q}$, because the inclusion (E.29) prescribes $\dot{\boldsymbol{q}} = \boldsymbol{0}$. Therefore, $\boldsymbol{v}(s)$ must be a solution to the associated impact differential inclusion $\dot{\boldsymbol{v}} \in D_{\boldsymbol{q}(0)}(\boldsymbol{v})$. Therefore, by Theorem 6,

$$\alpha_{\mathcal{Q}}(s_f) = 1 - \max_{[\boldsymbol{q}_f;\, \boldsymbol{v}_f] \in \mathcal{X}_f} \|\boldsymbol{v}_f\|_{\boldsymbol{M}(\boldsymbol{q}_f)} \in (0, 1]\,, \tag{E.32}$$

where the fact that $\mathcal{X}_f$ is closed implies the strict inequality $\alpha_{\mathcal{Q}}(s_f) > 0$. Setting $\alpha_{\mathcal{Q}}(0) = 0$ and selecting a configuration $\boldsymbol{q} \in \mathcal{Q}$, we now show that $\dot{\boldsymbol{v}} \in D_{\boldsymbol{q}}(\boldsymbol{v})$ is $\alpha_{\mathcal{Q}}(s)$-dissipative. Let $s_f > 0$, $\|\boldsymbol{v}_0\|_{\boldsymbol{M}(\boldsymbol{q})} = 1$, and $\boldsymbol{v}(s) \in \text{IVP}\,(D_{\boldsymbol{q}}, \boldsymbol{v}_0, [0, s_f])$. By Definition 2, $\boldsymbol{x}(s) = [\boldsymbol{q};\ \boldsymbol{v}(s)] \in \text{IVP}\,(D', \mathcal{X}_0, [0, s_f])$ and thus $\|\boldsymbol{v}(s)\|_{\boldsymbol{M}(\boldsymbol{q})} \leq 1 - \alpha_{\mathcal{Q}}(s) < 1$ for all $s \in [0, s_f]$.

E.4. Continuous-time Model Proofs

E.4.1. Proof of Theorem 7

Let $[a, b]$ and $\bar{\mathcal{X}}$ be compact. As $D(\bar{\boldsymbol{x}})$ neither depends on $t(\bar{\boldsymbol{x}})$ nor s, WLOG $[a, b] = [0, s_f]$ and $t(\bar{\boldsymbol{x}}) = 0$ for each $\bar{\boldsymbol{x}} \in \bar{\mathcal{X}}$. We will prove that IVP $(D, \bar{\boldsymbol{x}}, [0, s_f])$ has the claimed properties in the following manner:

1. We will bound kinetic energy growth (via Assumption 5), which will guarantee that solutions starting in $\bar{\mathcal{X}}$ remain in a larger open bounded set, $\bar{\mathcal{X}}'$.

2. We will show that, restricted to $\bar{\mathcal{X}}'$, $\dot{\bar{\boldsymbol{x}}} \in D(\bar{\boldsymbol{x}})$ is equivalent to another DI $\dot{\bar{\boldsymbol{x}}} \in \tilde{D}(\bar{\boldsymbol{x}})$ which is compatible with Proposition 2.

First, we construct a suitable $\bar{\mathcal{X}}'$. As $\bar{\mathcal{X}}$ is compact, we may pick $c > 0$ such that $\bar{\mathcal{X}} \subseteq \text{Ball}_c$. Let $\bar{\boldsymbol{x}}(s) \in \text{IVP}\left(D, \bar{\mathcal{X}}, [0, s_f]\right)$. We begin by establishing a bound on $\boldsymbol{v}(\bar{\boldsymbol{x}}(s))$ over $[0, s_f]$. Let

$K(\bar{\boldsymbol{x}}) = K(\boldsymbol{q}(\bar{\boldsymbol{x}}), \boldsymbol{v}(\bar{\boldsymbol{x}}))$. By Assumption 5, $\exists c_K > 0$ such that for all $\bar{\boldsymbol{x}}$,

$$\frac{\partial K}{\partial \bar{\boldsymbol{x}}} D_C(\bar{\boldsymbol{x}}) = \boldsymbol{v}^T F_s(\boldsymbol{x}, \mathcal{U}(\bar{\boldsymbol{x}})) \leq \sqrt{2} c_K \left\| \boldsymbol{v} \right\|_{\boldsymbol{M}} . \tag{E.33}$$

As the impact dynamics dissipate energy (Lemma 5),

$$\dot{K}(\bar{\boldsymbol{x}}(s)) \in \frac{\partial K}{\partial \bar{\boldsymbol{x}}} D(\bar{\boldsymbol{x}}) \leq 2 c_K \sqrt{K(\bar{\boldsymbol{x}})} . \tag{E.34}$$

Similar to the argument in Appendix E.2.2, we compare (E.34) to the differential equation $\dot{x} = 2c_K\sqrt{x}$, and upper bound K as

$$K(\boldsymbol{q}(s), \boldsymbol{v}(s)) \leq \left(\sqrt{K(\boldsymbol{q}(0), \boldsymbol{v}(0))} + c_K s \right)^2 . \tag{E.35}$$

Thus, picking $c_{\boldsymbol{M}}$ with $c_{\boldsymbol{M}}^{-1} \left\| \boldsymbol{v} \right\|_{\boldsymbol{M}} \leq \sqrt{2} \left\| \boldsymbol{v} \right\|_2 \leq c_{\boldsymbol{M}} \left\| \boldsymbol{v} \right\|_{\boldsymbol{M}}$,

$$\begin{aligned} \left\| \boldsymbol{v}(s) \right\|_2 &\leq c_{\boldsymbol{M}} \sqrt{K(\boldsymbol{q}(s), \boldsymbol{v}(s))} , & \text{(E.36)} \\ &\leq c_{\boldsymbol{M}} \left(\sqrt{K(\boldsymbol{q}(0), \boldsymbol{v}(0))} + c_K s \right) , & \text{(E.37)} \\ &\leq c_{\boldsymbol{M}}^2 \left\| \boldsymbol{v}(0) \right\|_2 + c_{\boldsymbol{M}} c_K s . & \text{(E.38)} \end{aligned}$$

Now, we bound $\boldsymbol{q}(\bar{\boldsymbol{x}}(s))$. As $\left\| \dot{\boldsymbol{q}} \right\|_2 \leq \left\| \boldsymbol{\Gamma} \right\|_F \left\| \boldsymbol{v} \right\|_2$, selecting $c_{\boldsymbol{\Gamma}} = \sup_{\boldsymbol{q}} \left\| \boldsymbol{\Gamma} \right\|_F$, we can apply the triangle inequality as:

$$\left\| \boldsymbol{q}(s) \right\|_2 \leq \left\| \boldsymbol{q}(0) \right\|_2 + c_{\boldsymbol{\Gamma}} s \max_{s' \in [0,s]} \left\| \boldsymbol{v}(s') \right\|_2 . \tag{E.39}$$

Finally, we bound $\left\| t(s) \right\| \leq s$ from $\dot{t} \leq 1$. As $\left\| \bar{\boldsymbol{x}}(0) \right\|_2 < c$,

$$\begin{aligned} \left\| \bar{\boldsymbol{x}}(s) \right\|_2 &\leq \left\| \boldsymbol{q}(s) \right\|_2 + \left\| \boldsymbol{v}(s) \right\|_2 + \left\| t(s) \right\| , & \text{(E.40)} \\ &< c + (c_{\boldsymbol{\Gamma}} s_f + 1) \left(c_{\boldsymbol{M}}^2 c + c_{\boldsymbol{M}} c_K s_f \right) + s_f . & \text{(E.41)} \end{aligned}$$

As (E.41) is constant, $\bar{\boldsymbol{x}}(s)$ remains in a bounded open set $\bar{\mathcal{X}}'$.

Now, we relate $\dot{\bar{\boldsymbol{x}}} \in D(\bar{\boldsymbol{x}})$ to another DI which can be analyzed via Proposition 2. First we show that

$D(\bar{\boldsymbol{x}})$ is u.s.c. For any $\boldsymbol{q}$ and separating velocity $\boldsymbol{v} \in \mathcal{S}(\boldsymbol{q})$, we can pick an open neighborhood $\mathcal{Q} \times \mathcal{V}$ of $[\boldsymbol{q};\ \boldsymbol{v}]$ which also consists solely of separating velocities by continuity of $\boldsymbol{\phi}$ and $\boldsymbol{J}_n$. Therefore, the set of separating-velocity states $\bar{\mathcal{X}}_{\mathcal{S}}$ is open. Furthermore, each of $D_{\mathcal{S}}(\bar{\boldsymbol{x}})$ (Assumption 4) and $D_{\mathcal{C}}(\bar{\boldsymbol{x}})$ (Corollary 1) are u.s.c.. $D(\bar{\boldsymbol{x}})$ must then be u.s.c., because it is constructed from two u.s.c. functions on disjoint open sets, and their convex hull on the remainder of the space.

By Assumption 6, $D(\bar{\mathcal{X}}')$ is bounded. As $\bar{\mathcal{X}}'$ is open and bounded, we can construct a bounded, non-empty, compact-convex valued u.s.c. function $\tilde{D}(\bar{\boldsymbol{x}})$ defined over $\mathbb{R}^n$ such that $\tilde{D}|_{\bar{\mathcal{X}}'} = D|_{\bar{\mathcal{X}}'}$. In particular,

$$\tilde{D}(\bar{\boldsymbol{x}}) = \begin{cases} D(\bar{\boldsymbol{x}}) & \bar{\boldsymbol{x}} \in \bar{\mathcal{X}}', \\ \text{co}\left(\text{cl}D(\bar{\mathcal{X}}')\right) & \text{otherwise}. \end{cases} \tag{E.42}$$

Therefore, by Theorem 15, IVP $\left(\tilde{D}, \bar{\boldsymbol{x}}, [0, s_f]\right)$ is non-empty, closed under uniform convergence, and u.s.c. on $\bar{\mathcal{X}}$. As D and $\tilde{D}$ are locally equivalent, IVP $\left(\tilde{D}, \bar{\boldsymbol{x}}, [0, s_f]\right) = \text{IVP}\left(D, \bar{\boldsymbol{x}}, [0, s_f]\right)$ on $\bar{\mathcal{X}}$ and the claim is proven.

E.4.2. Proof of Theorem 8

Suppose not. Then there exists a non-penetrating initial state $\bar{\boldsymbol{x}}_0 = [\boldsymbol{q}_0;\ \boldsymbol{v}_0;\ t_0] \notin \bar{\mathcal{X}}_P$, compact interval $[0, s_f]$, and corresponding solution $\bar{\boldsymbol{x}}(s) = [\boldsymbol{q}(s); \boldsymbol{v}(s); t(s)] \in \text{IVP}\left(D, \bar{\boldsymbol{x}}_0, [0, s_f]\right)$ that penetrates at some $s_P \in (0, s_f]$ ($\bar{\boldsymbol{x}}(s_P) \in \bar{\mathcal{X}}_P$). Thus some contact $i \in \mathcal{I}$ penetrates at s_P ($\boldsymbol{\phi}_i(\boldsymbol{q}(s_P)) < 0$). By the intermediate value theorem, we may select $s_A \in [0, s_P)$ such that $\boldsymbol{\phi}_i(\boldsymbol{q}(s_P)) < \boldsymbol{\phi}_i(\boldsymbol{q}(s_A)) < 0$ and contact i penetrates on the entire interval $[s_A, s_P]$. But then by the definition of $D(\bar{\boldsymbol{x}})$, $\boldsymbol{q}(s)$ and therefore $\boldsymbol{\phi}_i$ must be constant on $[s_A, s_P]$, a contradiction.

E.4.3. Proof of Theorem 9

Suppose not. Then there exists a compact interval $[a, b]$; solution $\bar{\boldsymbol{x}}(s) \in \text{SOL}\left(D, [a, b]\right)$ with $\bar{\boldsymbol{x}}(s)$ impacting but not penetrating, $\bar{\boldsymbol{x}}([a, b]) \subseteq \bar{\mathcal{X}}_{\mathcal{C}} \setminus \bar{\mathcal{X}}_P$; and set $\mathcal{S} = \left\{s : \dot{\bar{\boldsymbol{x}}}(s) \in D(\bar{\boldsymbol{x}}(s)) \setminus D_{\mathcal{C}}(\bar{\boldsymbol{x}}(s))\right\}$ with positive measure. Furthermore, $\dot{t}(s)|_{\mathcal{S}} > 0$ and $\dot{\boldsymbol{q}}(s) = \boldsymbol{\Gamma}(\boldsymbol{q}(s))\boldsymbol{v}(s)\dot{t}(s)$.

We will now show that allowing $\dot{t}(s)|_{\mathcal{S}} > 0$ must lead to penetration and therefore a contradiction with Theorem 8. By Lebesgue's density theorem, we may select a point of density $a < s_1 < b$, i.e.,

for all $\delta > 0$, $[s_1, s_1+\delta] \cap \mathcal{S}$ has non-zero measure. As $\bar{\boldsymbol{x}}(s)$ remains in $\bar{\mathcal{X}}_{\mathcal{C}}$, by continuity of $\boldsymbol{J}(\boldsymbol{q})$ and $\bar{\boldsymbol{x}}(s)$ we may select $\delta > 0$ and a contact i that is active $\boldsymbol{\phi}_i(\boldsymbol{q}(s)) = 0$ with negative time derivative $\boldsymbol{J}_{n,i}\boldsymbol{v}(s) < 0$ on $[s_1, s_1 + \delta] \subseteq [a, b]$. Let $\dot{\boldsymbol{\phi}}_{\max} = \max_{s\in[s_1,s_1+\delta]} \boldsymbol{J}_{n,i}\boldsymbol{v}(s) < 0$. Then

$$\boldsymbol{\phi}_i(s_1 + \delta) = \int_{[s_1,s_1+\delta]} \boldsymbol{J}_{n,i}\boldsymbol{v}(s)\dot{t}(s)\mathrm{d}s\,, \tag{E.43}$$

$$\leq \dot{\boldsymbol{\phi}}_{\max} \int_{[s_1,s_1+\delta]\cap\mathcal{S}} \dot{t}(s)\mathrm{d}s\,, \tag{E.44}$$

$$< 0\,, \tag{E.45}$$

and thus $\bar{\boldsymbol{x}}(s_1 + \delta) \in \bar{\mathcal{X}}_P$, a contradiction.

E.4.4. Proof of Theorem 10

Let $\bar{\mathcal{X}} \subseteq \bar{\mathcal{X}}_P^c$ be compact. By Corollary 1 there exists a dissipation rate $\alpha_{\bar{\mathcal{X}}}(s)$ such that the impact differential inclusion $\dot{\boldsymbol{v}} \in D_{\boldsymbol{q}}(\boldsymbol{v})$ for each configuration $\boldsymbol{q} \in \boldsymbol{q}(\bar{\mathcal{X}})$ is $\alpha_{\bar{\mathcal{X}}}(s)$-dissipative. Let $\bar{K} = \max_{\bar{\mathcal{X}}} \|\boldsymbol{v}\|_{\boldsymbol{M}(\boldsymbol{q})}$.

Suppose the claim is not true. Then, for some $s_f > s^*(\bar{\mathcal{X}}) = \frac{\bar{K}}{\alpha_{\bar{X}}(1)}$, there must exist a sequence of solutions $(\bar{\boldsymbol{x}}_j(s))_{j\in\mathbb{N}}$, $\bar{\boldsymbol{x}}_j(s) \in \mathrm{IVP}\left(D, \bar{\mathcal{X}}, [0, s_f]\right)$, for which the elapsed times grows arbitrarily small: $t_j(s_f) - t_j(0) \to 0$. By Theorem 7, $\mathrm{IVP}\left(D, \bar{\mathcal{X}}, [0, s_f]\right)$ is compact, and therefore by Assumption 6, the derivatives $\dot{\bar{\boldsymbol{x}}}_j(s)$ are uniformly bounded. Therefore $(\bar{\boldsymbol{x}}_j(s))_{j\in\mathbb{N}}$ is equicontinuous. Thus by Theorem 15, a subsequence of $\bar{\boldsymbol{x}}_j(s)$ converges uniformly to some $\bar{\boldsymbol{x}}_\infty(s)$ with $t_\infty([0, s_f]) = t_\infty(0)$. As $\mathrm{IVP}\left(D, \bar{\mathcal{X}}, [0, s_f]\right)$ is closed (Theorem 7), $\bar{\boldsymbol{x}}_\infty(s)$ must also solve the initial value problem.

We now show a contradiction arises because $\bar{\boldsymbol{x}}_\infty(s)$ follows impact dynamics longer than $\frac{\bar{K}}{\alpha_{\bar{X}}(1)}$. As $t_\infty(s)$ is constant, $\dot{t}_\infty(s) = 0$, and thus $\bar{\boldsymbol{x}}_\infty(s)$ is following only impact dynamics, $\bar{\boldsymbol{x}}_\infty(s) \in \mathrm{IVP}\left(D_{\mathcal{C}}, \bar{\mathcal{X}}, [0, s_f]\right)$. In order for $\dot{\bar{\boldsymbol{x}}}_\infty(s)$ to be selected from $D_{\mathcal{C}}$, we must have $\bar{\boldsymbol{x}}_\infty(s) \notin \bar{\mathcal{X}}_{\mathcal{S}}$ a.e., and thus $\boldsymbol{v}_\infty(s) \notin \mathcal{S}(\boldsymbol{q}_\infty(s))$ a.e. Additionally, as $\bar{\boldsymbol{x}}_\infty(s)$ only follows impact dynamics, the configuration is constant, i.e. $\boldsymbol{q}_\infty([0, s_f]) = \boldsymbol{q}_\infty(0) = \boldsymbol{q}_\infty$. Therefore $\boldsymbol{v}_\infty(s)$ is a solution of $\dot{\boldsymbol{v}} \in D_{\boldsymbol{q}_\infty}(\boldsymbol{v})$, and $\boldsymbol{v}_\infty(s) \in \mathrm{cl}\mathcal{C}(\boldsymbol{q}_\infty)$. Therefore $\mathcal{C}(\boldsymbol{q}_\infty)$ is non-empty and therefore has active contact ($\boldsymbol{q}_\infty \in \mathcal{Q}_A$). As, as $\bar{\mathcal{X}}$ is closed, $\boldsymbol{q}_\infty \in \boldsymbol{q}(\bar{\mathcal{X}})$ and thus $\dot{\boldsymbol{v}} \in D_{\boldsymbol{q}_\infty}(\boldsymbol{v})$ is $\alpha_{\bar{\mathcal{X}}}(s)$-dissipative. Finally, by Lemma 6,

$s_f < \frac{\|\boldsymbol{v}_\infty(0)\|_{\boldsymbol{M}}}{\alpha_{\bar{\mathcal{X}}}(1)} \leq s^*(\bar{\mathcal{X}})$, a contradiction.

E.4.5. Proof of Corollary 2

As $\bar{\mathcal{X}}(s_f)$ is non-empty and compact for all $s_f > 0$, $t_f(s_f)$ is well-defined. Then, $\liminf_{s_f \to \infty} \frac{t_f(s_f)}{s_f} \in [0,1]$ as the DI (7.19) enforces $\dot{t}(s) \in [0,1]$. Consider a particular $s_f > 0$, and let $\bar{\boldsymbol{x}}(s) \in \text{IVP}\left(D, \bar{\mathcal{X}}, [0, s_f]\right)$. By Theorem 10, $t(s)$ increases by $t^*(\bar{\mathcal{X}})$ over each interval of duration $s^*(\bar{\mathcal{X}})$, bounding

$$\frac{t_f(s_f)}{s_f} \geq \frac{t^*(\bar{\mathcal{X}})}{s_f} \left\lfloor \frac{s_f}{s^*(\bar{\mathcal{X}})} \right\rfloor \geq \frac{t^*(\bar{\mathcal{X}})}{s^*(\bar{\mathcal{X}})} - \frac{t^*(\bar{\mathcal{X}})}{s_f}\,. \tag{E.46}$$

Therefore, $\liminf_{s_f \to \infty} \frac{t_f(s_f)}{s_f} \geq \frac{t^*(\bar{\mathcal{X}})}{s^*(\bar{\mathcal{X}})}$.

E.5. Simulation Proofs

E.5.1. Proof of Theorem 11

Consider some state $[\boldsymbol{q};\ \boldsymbol{v}]$ and normal impulse $\boldsymbol{\lambda}_{n,max} \geq \boldsymbol{0}$. Let $\boldsymbol{z} = [\boldsymbol{\beta};\ \bar{\boldsymbol{\lambda}};\ \boldsymbol{\gamma}]$ Then we have

$$\boldsymbol{z}^T \boldsymbol{W}_{\boldsymbol{q}} \boldsymbol{z} = \frac{1}{2} \boldsymbol{z}^T \left(\boldsymbol{W}_{\boldsymbol{q}} + \boldsymbol{W}_{\boldsymbol{q}}^T \right) \boldsymbol{z}\,, \tag{E.47}$$

$$= \left\| \bar{\boldsymbol{J}}^T \bar{\boldsymbol{\lambda}} \right\|^2_{\boldsymbol{M}^{-1}} + \boldsymbol{\lambda}_n^T \boldsymbol{\mu} \boldsymbol{\gamma}\,, \tag{E.48}$$

$$\geq 0\,, \tag{E.49}$$

where the final inequality holds because $\boldsymbol{\mu}$ has positive entries and $\boldsymbol{M} \succ 0$. Therefore, $\boldsymbol{M}_{\boldsymbol{q}}$ is copositive.

Suppose further that $\boldsymbol{z} \in \text{LCP}(\boldsymbol{W}_{\boldsymbol{q}}, \boldsymbol{0})$, thus $\boldsymbol{W}_{\boldsymbol{q}} \boldsymbol{z} \geq \boldsymbol{0}$ and $\boldsymbol{z}^T \boldsymbol{W}_{\boldsymbol{q}} \boldsymbol{z} = 0$. $\boldsymbol{W}_{\boldsymbol{q}} \boldsymbol{z} \geq \boldsymbol{0}$ implies by construction that

$$\boldsymbol{\lambda}_n \leq \boldsymbol{0}\,, \qquad\qquad \boldsymbol{E}^T \boldsymbol{\lambda}_D \leq \boldsymbol{\mu} \boldsymbol{\lambda}_n \leq \boldsymbol{0}\,. \tag{E.50}$$

Therefore as $\boldsymbol{\lambda}_n, \boldsymbol{\lambda}_D \geq \boldsymbol{0}$, $\boldsymbol{\lambda}_n = \boldsymbol{0}$ and $\boldsymbol{\lambda}_D = \boldsymbol{0}$. Finally, as $\boldsymbol{\lambda}_{n,max}$ and $\boldsymbol{\beta}$ are non-negative,

$$\boldsymbol{z}^T \boldsymbol{w}_{\boldsymbol{q}}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max}) = \boldsymbol{\beta}^T \boldsymbol{\lambda}_{n,max} \geq \boldsymbol{0}\,. \tag{E.51}$$

Therefore by Proposition 3, LCP($\boldsymbol{W_q}, \boldsymbol{w_q}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max})$) is non-empty.

E.5.2. Proof of Theorem 12

Consider a state $[\boldsymbol{q};\ \boldsymbol{v}]$, normal impulse increment $\boldsymbol{\lambda}_{n,max} \geq \boldsymbol{0}$, and solution to the impact LCP $\boldsymbol{z} = \left[\boldsymbol{\beta};\ \bar{\boldsymbol{\lambda}};\ \boldsymbol{\gamma}\right] \in \text{LCP}(\boldsymbol{W_q}, \boldsymbol{w_q}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max}))$. Let $\boldsymbol{v}' = \boldsymbol{v} + \boldsymbol{M}^{-1}\bar{\boldsymbol{J}}^T\bar{\boldsymbol{\lambda}}$. Then from the complementarity condition we have

$$
\begin{aligned}
0 &= \boldsymbol{z}^T \left(\boldsymbol{W_q}\boldsymbol{z} + \boldsymbol{w_q}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max})\right) , && \text{(E.52)}\\
&= \left(\bar{\boldsymbol{\lambda}}^T\bar{\boldsymbol{J}}\right)\boldsymbol{v}' + \boldsymbol{\lambda}_n^T\boldsymbol{\mu}\boldsymbol{\gamma} + \boldsymbol{\beta}^T\boldsymbol{\lambda}_{n,max} , && \text{(E.53)}\\
&= \left(\boldsymbol{v}' - \boldsymbol{v}\right)^T \boldsymbol{M}\boldsymbol{v}' + \boldsymbol{\lambda}_n^T\boldsymbol{\mu}\boldsymbol{\gamma} + \boldsymbol{\beta}^T\boldsymbol{\lambda}_{n,max} , && \text{(E.54)}\\
&= \left\|\boldsymbol{v}'\right\|_{\boldsymbol{M}}^2 - \boldsymbol{v}^T\boldsymbol{M}\boldsymbol{v}' + \boldsymbol{\lambda}_n^T\boldsymbol{\mu}\boldsymbol{\gamma} + \boldsymbol{\beta}^T\boldsymbol{\lambda}_{n,max} , && \text{(E.55)}
\end{aligned}
$$

As $\boldsymbol{\lambda}_n^T\boldsymbol{\mu}\boldsymbol{\gamma} + \boldsymbol{\beta}^T\boldsymbol{\lambda}_{n,max} \geq 0$, $\|\boldsymbol{v}'\|_{\boldsymbol{M}}^2 \leq \boldsymbol{v}^T\boldsymbol{M}\boldsymbol{v}'$. Cauchy-Schwartz then gives $\|\boldsymbol{v}'\|_{\boldsymbol{M}}^2 \leq \|\boldsymbol{v}\|_{\boldsymbol{M}} \|\boldsymbol{v}'\|_{\boldsymbol{M}}$, and thus $K(\boldsymbol{q}, \boldsymbol{v}') - K(\boldsymbol{q}, \boldsymbol{v}) \leq 0$.

E.5.3. Proof of Lemma 7

Let $[\boldsymbol{q};\ \boldsymbol{v}]$ be an impacting state ($\boldsymbol{v} \in \mathcal{C}(\boldsymbol{q})$), and let $\boldsymbol{\lambda}_{n,max} > \boldsymbol{0}$ be a normal impulse. Consider an impact LCP solution

$$
[\boldsymbol{\beta};\ \boldsymbol{\lambda}_n;\ \boldsymbol{\lambda}_D;\ \boldsymbol{\gamma}] \in \text{LCP}(\boldsymbol{W_q}, \boldsymbol{w_q}(\boldsymbol{v}, \boldsymbol{\lambda}_{n,max})) .
$$

such that

$$
\boldsymbol{\lambda}_n < \boldsymbol{\lambda}_{n,max} . \tag{E.56}
$$

Therefore for each contact i, the complementary equation (7.24) yields $\boldsymbol{\beta}_i = 0$ as $\boldsymbol{\lambda}_{n,max_i} - \boldsymbol{\lambda}_{n,i} > 0$. Then from complementarity equation (7.25), $\boldsymbol{J}_{n,i}\boldsymbol{v}' \geq 0$.

E.5.4. Proof of Lemma 8

Consider a configuration $\boldsymbol{q} \in \mathcal{Q}_A \setminus \mathcal{Q}_P$ and $\bar{\boldsymbol{\lambda}} = [\boldsymbol{\lambda}_n;\ \boldsymbol{\lambda}_D]$ obeying (2.31). By construction, $\bar{\boldsymbol{J}}^T\bar{\boldsymbol{\lambda}} \in \text{LFC}(\boldsymbol{q})$. Let

$$
\mathcal{F} = \text{LFC}\left(\boldsymbol{q}\right) \cap \bar{\boldsymbol{J}}^T \left\{[\boldsymbol{\lambda}_n;\ \boldsymbol{\lambda}_D] : \|\boldsymbol{\lambda}_n\|_1 = 1\right\} . \tag{E.57}
$$

As $\text{LFC}(\boldsymbol{q})$ is a convex cone, $\boldsymbol{r}$ satisfies the claim if $\mathcal{F} \cdot \boldsymbol{M}^{-1}\boldsymbol{r} > 1$. As $\text{LFC} \subseteq \text{FC}$, by Assumption 2, $\mathbf{0} \notin \mathcal{F}$. $\mathcal{F}$ is compact, non-empty, and convex polyhedron. Therefore, by **?**, Theorem 11.4 there exists $\tilde{\boldsymbol{r}}$ such that

$$\varepsilon = \min_{\boldsymbol{F} \in \mathcal{F}} \boldsymbol{F} \cdot \tilde{\boldsymbol{r}} > \max_{\boldsymbol{F} \in -\mathcal{F}} \boldsymbol{F} \cdot \tilde{\boldsymbol{r}} = -\varepsilon \,. \tag{E.58}$$

Setting $\boldsymbol{r}(\boldsymbol{q}) = \frac{\boldsymbol{M}(\boldsymbol{q})\tilde{\boldsymbol{r}}}{\varepsilon}$ satisfies the claim.

E.5.5. Proof of Theorem 13

Let $\boldsymbol{q}_0 \in \mathcal{Q}_A \setminus \mathcal{Q}_P$ be a pre-impact configuration; let $\boldsymbol{v}_0 \in \mathcal{C}(\boldsymbol{q}_0)$ be a pre-impact velocity; and let $h > 0$ be a step size. As each $\boldsymbol{\lambda}_{n,max}$ is selected from the uniform distribution over the h-width box, we have that

$$c_p = \mathbb{E}_{\boldsymbol{\lambda}_{n,max} \sim h \cdot p} \left[\min_i \boldsymbol{\lambda}_{n,max_i} \right] = \frac{h}{m+1} \,. \tag{E.59}$$

We assume WLOG that p is supported on the interior of the unit box $(0,1)^m$, as the probability of being on the boundary is 0. Let $\sigma = \sigma_{min}(\boldsymbol{M} = \boldsymbol{M}(\boldsymbol{q}_0))$, and therefore $\sqrt{\sigma}\,\|\boldsymbol{v}\|_2 \leq \|\boldsymbol{v}\|_{\boldsymbol{M}}$. Now, select $\boldsymbol{r}$ for $\boldsymbol{q}_0$ as defined in Lemma 8. We will now show that the existence of $\boldsymbol{r}$ in conjunction with dissipation (Theorem 12), allows us to create a useful sufficient condition for impact termination.

Consider any execution of Algorithm 1 with initial state $[\boldsymbol{q}_0;\ \boldsymbol{v}_0]$, and let $\boldsymbol{\lambda}^k_{n,max}$, $\bar{\boldsymbol{\lambda}}^k = [\boldsymbol{\lambda}^k_n;\ \boldsymbol{\lambda}^k_D]$ and $\boldsymbol{v}_k$ be the maximum normal impulse; selected impulse; and velocity computed on lines 3–5 on the kth iteration of the loop. If the loop has not terminated after K steps, then for all loop iterations

$k \in \{1, \ldots, K\}$, $\boldsymbol{v}_k \in \mathcal{C}(\boldsymbol{q}_0)$. By Theorem 12 and Lemmas 7–8, we have that

$$\begin{aligned}
\|\boldsymbol{v}_0\|_{\boldsymbol{M}} &\geq \|\boldsymbol{v}_K\|_{\boldsymbol{M}} \,, && \text{(E.60)}\\
&\geq \sqrt{\sigma}\, \|\boldsymbol{v}_K\|_2 \,, && \text{(E.61)}\\
&\geq \sqrt{\sigma} \frac{\boldsymbol{r}}{\|\boldsymbol{r}\|_2} \cdot \boldsymbol{v}_K && \text{(E.62)}\\
&\geq \frac{\sqrt{\sigma}}{\|\boldsymbol{r}\|_2} \left(\boldsymbol{r} \cdot \boldsymbol{v}_0 + \sum_{k=1}^{K} \left\| \boldsymbol{\lambda}_n^k \right\|_1 \right) \,, && \text{(E.63)}\\
&\geq -\sqrt{\sigma}\, \|\boldsymbol{v}_0\|_2 + \frac{\sqrt{\sigma}}{\|\boldsymbol{r}\|_2} \sum_{k=1}^{K} \left\| \boldsymbol{\lambda}_n^k \right\|_1 \,, && \text{(E.64)}\\
&\geq -\|\boldsymbol{v}_0\|_{\boldsymbol{M}} + \sum_{k=1}^{K} \frac{\sqrt{\sigma}}{\|\boldsymbol{r}\|_2} \min_i \boldsymbol{\lambda}_{n,max_i}^k \,. && \text{(E.65)}
\end{aligned}$$

For this inequality to hold, and thus for $\boldsymbol{v}_K$ to remain in $\mathcal{C}(\boldsymbol{q}_0)$, it must be true that the summation in (E.65) is no greater than $2\|\boldsymbol{v}_0\|_{\boldsymbol{M}}$. Therefore, termination of the impact within K steps (i.e. $Z(h, \boldsymbol{q}_0, \boldsymbol{v}_0) \leq K$) is implied by $Z_K > c_Z \|\boldsymbol{v}_0\|_{\boldsymbol{M}}$, where

$$\begin{aligned}
c_Z &= \frac{2\|\boldsymbol{r}\|_2}{\sqrt{\sigma}} \,, && \text{(E.66)}\\
Z_K &= \sum_{k=1}^{K} \min_i \boldsymbol{\lambda}_{n,max_i}^k \,. && \text{(E.67)}
\end{aligned}$$

Given that the $\boldsymbol{\lambda}_{n,max} \sim h \cdot p$ are selected i.i.d. we have that $\mathbb{E}[Z_K] = K c_p$. Thus we would expect an impact to terminate proportional to

$$K^* = \left\lceil \frac{c_Z}{c_p} \right\rceil \lceil \|\boldsymbol{v}_0\|_{\boldsymbol{M}} \rceil \,. \tag{E.68}$$

We now bound the termination time Z using Hoeffding's inequality, applied below in (E.72); for

$k \in \mathbb{Z}^+$ and $K = 2K^* + k$,

$$P(Z \geq K) \leq P\left(Z_K \leq c_Z \left\|\boldsymbol{v}_0\right\|_{\boldsymbol{M}}\right), \tag{E.69}$$

$$\leq P\left(Z_K \leq K^* c_p\right), \tag{E.70}$$

$$= P\left(Z_K - K c_p \leq -\left(K^* + k\right) c_p\right), \tag{E.71}$$

$$\leq \exp\left(-\frac{2}{K}\left(K^* + k\right)^2 \frac{c_p^2}{h^2}\right), \tag{E.72}$$

$$\leq \exp\left(-\left(K^* + k\right) \frac{c_p^2}{h^2}\right), \tag{E.73}$$

$$\leq \exp\left(-\frac{k}{(m+1)^2}\right). \tag{E.74}$$

Thus the claim is satisfied.

E.5.6. Proof of Lemma 11

Suppose not. Then there exists a configuration $\boldsymbol{q} \in \mathcal{Q}_A \setminus \mathcal{Q}_P$, velocity $\boldsymbol{v}$, and $\varepsilon > 0$, such that for all $N \in \mathbb{N}$, there exists a $\boldsymbol{v}_N$, $\boldsymbol{J}_n \boldsymbol{v}_N \geq -\frac{1}{N}$, $\left\|\boldsymbol{v}_N\right\|_{\boldsymbol{M}} \leq \left\|\boldsymbol{v}\right\|_{\boldsymbol{M}}$, and yet $\boldsymbol{v}'_N = \boldsymbol{f}_{\boldsymbol{q}}(\boldsymbol{v}_N, \varepsilon \boldsymbol{1}) \in \mathcal{C}(\boldsymbol{q})$.

Due to energy dissipation (Theorem 12) and the boundedness of $\boldsymbol{v}_N$, the sequence $\boldsymbol{v}'_N$ is bounded as well. Without loss of generality we can therefore assume that $\boldsymbol{v}_N \to \boldsymbol{v}_\infty$ and $\boldsymbol{v}'_N \to \boldsymbol{v}'_\infty$. As $\boldsymbol{J}_n \boldsymbol{v}_N \geq -\frac{1}{N}$, it must be that $\boldsymbol{J}_n \boldsymbol{v}_\infty \geq 0$. Therefore, $\boldsymbol{v}'_\infty = \boldsymbol{f}_{\boldsymbol{q}}(\boldsymbol{v}_\infty, \varepsilon \boldsymbol{1}_m) = \boldsymbol{v}_\infty$ via Lemma 10. As $\boldsymbol{v}_N$ and $\boldsymbol{v}'_N$ converge to each other, there exists an N^*, with LCP-selected force $\bar{\boldsymbol{\lambda}}_{N^*} = [\boldsymbol{\lambda}_n;\ \boldsymbol{\lambda}_D]$ such that

$$\left\|\left(\boldsymbol{v}'_{N^*} - \boldsymbol{v}_{N^*}\right)\right\|_2 = \left\|\boldsymbol{M}^{-1} \bar{\boldsymbol{J}}^T \bar{\boldsymbol{\lambda}}_{N^*}\right\|_2 < \frac{\varepsilon}{\left\|\boldsymbol{r}(\boldsymbol{q})\right\|_2}, \tag{E.75}$$

where $\boldsymbol{r}(\boldsymbol{q})$ comes from Lemma 8. However, by Lemma 7, as $\boldsymbol{v}'_{N^*} \in \mathcal{C}(\boldsymbol{q})$, at least one contact must fully activate, and thus $\left\|\boldsymbol{\lambda}_n\right\|_1 \geq \varepsilon$. But then again by Lemma 8, $\left\|\boldsymbol{M}^{-1} \bar{\boldsymbol{J}}^T \bar{\boldsymbol{\lambda}}_{N^*}\right\|_2 \geq \frac{\varepsilon}{\left\|\boldsymbol{r}(\boldsymbol{q})\right\|_2}$, a contradiction.

E.5.7. Proof of Theorem 14

First we show that generating an ε-net of $\mathcal{V}_\infty(\boldsymbol{x}_0, h) \setminus \mathcal{C}(\boldsymbol{q}_0)$ can be reduced to generating an ε'-net of $\mathcal{V}_N(\boldsymbol{x}_0, h)$ for a suitable (ε', N). We then show that $\mathcal{V}_N(\boldsymbol{x}_0, h)$ is the image of a box under a

Lipschitz function, and apply Proposition 5.

Select an initial $\boldsymbol{x}_0 = [\boldsymbol{q}_0;\ \boldsymbol{v}_0] \in (\mathcal{Q}_A \setminus \mathcal{Q}_P) \times \mathbb{R}^{n_v}$; step size $h > 0$; and constants $\varepsilon, \delta > 0$. Define ψ as on line 2 of Alg. 2. Select

$$\varepsilon' = \min\left(\frac{\varepsilon}{3}, \frac{\delta\left(\frac{\varepsilon}{3\psi}, \boldsymbol{v}_0\right)}{2\sigma_{max}(\boldsymbol{J}_n)}\right), \tag{E.76}$$

where $\delta\left(\frac{\varepsilon}{3\psi}, \boldsymbol{v}_0\right)$ comes from Lemma 11. Via Lemma 12, select N such that $\mathcal{V}_N(\boldsymbol{x}_0, h)$ is an ε'-net of $\mathcal{V}_\infty(\boldsymbol{x}_0, h)$. Consider a run of Approximate$(h, \boldsymbol{x}_0, \epsilon, N, M)$ for some $M > 0$. Suppose that the M samples generated on line 4 of Alg. 2 constitute a ε' net of $\mathcal{V}_N(\boldsymbol{x}_0, h)$. Consider a post-impact velocity $\boldsymbol{v}_1 \in \mathcal{V}_\infty(\boldsymbol{x}_0, h) \setminus \mathcal{C}(\boldsymbol{q}_0)$. Then there exists a $\boldsymbol{v}_2 \in \mathcal{V}_N(\boldsymbol{x}_0, h)$ with $\|\boldsymbol{v}_1 - \boldsymbol{v}_2\|_2 < \varepsilon'$. Pick the closest $\boldsymbol{v}_3$ to $\boldsymbol{v}_2$ selected on line 4 of Alg. 2. From (E.76), we know that $\|\boldsymbol{v}_3 - \boldsymbol{v}_2\|_2 \leq \frac{\varepsilon}{3}$, and $\boldsymbol{J}_n\boldsymbol{v}_3 \geq -\delta\left(\frac{\varepsilon}{3\psi}, \boldsymbol{v}_0\right)$. $\boldsymbol{v}_3$ is used to generate $\boldsymbol{v}_4 = \boldsymbol{f}_{\boldsymbol{q}_0}(\boldsymbol{v}_3, \frac{\varepsilon}{3\psi}\mathbf{1}_m)$ on line 5, and $\boldsymbol{J}_n\boldsymbol{v}_4 \geq \mathbf{0}$ via Lemma 11.

$\boldsymbol{v}_4$ is thus in the post-impact set $\mathcal{V}_\infty(\boldsymbol{x}_0, h) \setminus \mathcal{C}(\boldsymbol{q}_0)$ and is output by Approximate$(h, \boldsymbol{x}_0, \epsilon, N, M)$. Suppose that $\bar{\boldsymbol{\lambda}} = [\boldsymbol{\lambda}_n;\ \boldsymbol{\lambda}_D]$ was the LCP-selected force in the calculation of $\boldsymbol{v}_4$; we then have that $\|\boldsymbol{v}_4 - \boldsymbol{v}_3\|_2 \leq \frac{\varepsilon}{3}$ by construction of ψ on line 2 of Algorithm 2. Thus, $\|\boldsymbol{v}_4 - \boldsymbol{v}_1\|_2$ is smaller than

$$\|\boldsymbol{v}_2 - \boldsymbol{v}_1\|_2 + \|\boldsymbol{v}_3 - \boldsymbol{v}_2\|_2 + \|\boldsymbol{v}_4 - \boldsymbol{v}_3\|_2 \leq \varepsilon\,. \tag{E.77}$$

Therefore, the claim is true if the samples from $\mathcal{V}_N(\boldsymbol{x}_0, h)$ generated on line 4 of Algorithm 2 are a ε' net of $\mathcal{V}_N(\boldsymbol{x}_0, h)$ with probability $1 - \delta$; we now calculate a M that guarantees this property.

Consider the sequence of functions

$$\boldsymbol{f}^1(\boldsymbol{\lambda}_n^1) = \boldsymbol{f}_{\boldsymbol{q}_0}(\boldsymbol{v}_0, \boldsymbol{\lambda}_n^1)\,, \tag{E.78}$$

$$\boldsymbol{f}^k(\boldsymbol{\lambda}_n^1, \ldots, \boldsymbol{\lambda}_n^k) = \boldsymbol{f}_{\boldsymbol{q}_0}(\boldsymbol{f}^{k-1}(\boldsymbol{\lambda}_n^1, \ldots, \boldsymbol{\lambda}_n^{k-1}), \boldsymbol{\lambda}_n)\,. \tag{E.79}$$

Examining (7.35), we see that

$$\mathcal{V}_N(\boldsymbol{x}_0, h) = \boldsymbol{f}^N([0, h]^{Nm})\,. \tag{E.80}$$

Furthermore, if $f_{\boldsymbol{q}_0}$ has Lipschitz constant L, then $\boldsymbol{f}^N$ is Lipschitz with constant L^N by Proposition 1. Under Assumption 7, $\mathrm{Sim}(h, \boldsymbol{x}_0, N)$ yields a uniform sample of $[0,h]^{Nm}$ mapped under $\boldsymbol{f}^N$. Therefore, the claim holds, with M given by Proposition 5:

$$M \geq \frac{\ln(\delta\Omega)}{\ln(1-\Omega)}\,, \qquad\qquad \Omega = \left\lceil \frac{hL^N\sqrt{Nm}}{\varepsilon'} \right\rceil^{-Nm}\,. \tag{E.81}$$

BIBLIOGRAPHY

R. J. Adcock. A problem in least squares. *Analyst*, 5:53, 1878.

S. N. Afriat. Theory of maxima and the method of lagrange. *SIAM Journal on Applied Mathematics*, 20(3):343–357, 1971. doi: 10.1137/0120037. URL https://doi.org/10.1137/0120037.

A. Agrawal, S. Barratt, S. Boyd, E. Busseti, and W. Moursi. Differentiating through a cone program. *Journal of Applied and Numerical Optimization*, 1(2):107–115, 2019.

A. Ajay, J. Wu, N. Fazeli, M. Bauza, L. P. Kaelbling, J. B. Tenenbaum, and A. Rodriguez. Augmenting physical simulators with stochastic neural networks: Case study of planar pushing and bouncing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018. doi: 10.1109/IROS.2018.8593995.

Anurag Ajay, Maria Bauza, Jiajun Wu, Nima Fazeli, Joshua B Tenenbaum, Alberto Rodriguez, and Leslie P Kaelbling. Combining physical simulators and object-based networks for control. In *International Conference on Robotics and Automation (ICRA)*, 2019.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2623–2631, New York, NY, USA, 2019. Association for Computing Machinery. doi: 10.1145/3292500.3330701.

Cannarsa Piermarco Albano, Paolo. Structural properties of singularities of semiconcave functions. *Annali della Scuola Normale Superiore di Pisa - Classe di Scienze*, 28(4):719–740, 1999. URL http://eudml.org/doc/84395.

Kelsey R Allen, Tatiana Lopez Guevara, Yulia Rubanova, Kim Stachenfeld, Alvaro Sanchez-Gonzalez, Peter Battaglia, and Tobias Pfaff. Graph network simulators can learn discontinuous, rigid contact dynamics. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=rbIzq-I84i_.

Kelsey R Allen, Yulia Rubanova, Tatiana Lopez-Guevara, William F Whitney, Alvaro Sanchez-Gonzalez, Peter Battaglia, and Tobias Pfaff. Learning rigid dynamics with face interaction graph networks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=J7Uh781A05p.

Brandon Amos, Lei Xu, and J. Zico Kolter. Input convex neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 146–155. PMLR, 2017.

M Anitescu and F A Potra. Formulating dynamic multi-rigid-body contact problems with friction

as solvable linear complementarity problems. *Nonlinear Dynamics*, 14(3):231–247, 1997.

Mihai Anitescu. Optimization-based simulation of nonsmooth rigid multibody dynamics. *Mathematical Programming*, 105(1):113–143, April 2005. doi: 10.1007/s10107-005-0590-7. URL https://doi.org/10.1007/s10107-005-0590-7.

Rika Antonova, Jingyun Yang, Krishna Murthy Jatavallabhula, and Jeannette Bohg. Rethinking optimization with differentiable simulation from a global perspective. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 276–286. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/antonova23a.html.

Jean-Pierre Aubin and Arrigo Cellina. *Differential Inclusions: Set-Valued Maps and Viability Theory (Grundlehren der mathematischen wissenschaften)*. Springer Verlag, 1984. ISBN 0387131051.

Alp Aydinoglu, Philip Sieg, Victor Preciado, and Michael Posa. Stabilization of complementarity systems via contact-aware controllers. *IEEE Transactions on Robotics (TRO)*, 38(3):1735–1754, 2022. doi: 10.1109/TRO.2021.3120931.

Alp Aydinoglu, Adam Wei, and Michael Posa. Consensus complementarity control for multi-contact mpc. *arXiv preprint arXiv:2304.11259*, April 2023.

David Baraff. Coping with friction for non-penetrating rigid body simulation. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, pages 31–41, New York, NY, USA, 1991. Association for Computing Machinery. doi: 10.1145/122718.122722.

C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. 22(4):469–483, dec 1996. doi: 10.1145/235815.235821.

Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and koray kavukcuoglu. Interaction networks for learning about objects, relations and physics. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/3147da8ab4a0437c15ef51a5cc7f2dc4-Paper.pdf.

J. Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 1(1):1–16, 1972. ISSN 0045-7825.

M S Bazaraa, Hanif D Sherali, and C M Shetty. *Nonlinear programming*. John Wiley & Sons, Nashville, TN, 3 edition, April 2006.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. doi: 10.1073/pnas.1903070116.

Dimitri Bertsekas. *Nonlinear programming*. Athena Scientific, September 2016.

Bibit Bianchini, Mathew Halm, Nikolai Matni, and Michael Posa. Generalization bounded implicit learning of nearly discontinuous functions. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference (L4DC)*, volume 168 of *Proceedings of Machine Learning Research*, pages 1112–1124. PMLR, 23–24 Jun 2022. URL https://proceedings.mlr.press/v168/bianchini22a.html.

Bibit Bianchini, Mathew Halm, and Michael Posa. Simultaneous learning of contact and continuous dynamics. *Under Review*, 2023.

Jérôme Bolte, Aris Danilidis, Olivier Ley, and Laurent Mazet. Characterizations of lojasiewicz inequalities: Subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society*, 362(6):3319–3363, 2010. ISSN 00029947. URL http://www.jstor.org/stable/25677828.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN 9780521833783.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

B Brogliato. *Nonsmooth mechanics: models, dynamics, and control*. Springer Verlag, 1999.

Samuel A. Burden, S. Shankar Sastry, Daniel E. Koditschek, and Shai Revzen. Event–selected vector field discontinuities yield piecewise–differentiable flows. *SIAM Journal on Applied Dynamical Systems*, 15(2):1227–1267, jan 2016. ISSN 1536-0040.

Anindya Chatterjee. *Rigid body collisions: some general considerations, new collision laws, and some experimental data*. PhD thesis, Cornell University, 7 1997.

Anindya Chatterjee. On the realism of complementarity conditions in rigid body collisions. *Nonlinear Dynamics*, 20(2):159–168, 1999. doi: 10.1023/a:1008397905242.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pages 4754–4765, 2018.

Giulio Cifarelli. Measurement error models. *Journal of Applied Econometrics*, 3(4):315–317, 1988. ISSN 08837252, 10991255. URL http://www.jstor.org/stable/2096647.

Simon Le Cleac'h, Hong-Xing Yu, Michelle Guo, Taylor A. Howell, Ruohan Gao, Jiajun Wu, Zachary

Manchester, and Mac Schwager. Differentiable physics simulation of dynamics-augmented neural objects. *IEEE Robotics Autom. Lett.*, 8(5):2780–2787, 2023.

Richard W. Cottle, Jong-Shi Pang, and Richard E. Stone. *The Linear Complementarity Problem*. SIAM, 2009.

Erwin Coumans. Bullet physics simulation. In *ACM SIGGRAPH 2015 Courses*, SIGGRAPH '15. ACM, 2015.

Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Steven P. Dirkse and Michael C. Ferris. The path solver: a nonmonotone stabilization scheme for mixed complementarity problems. *Optimization Methods and Software*, 5(2):123–156, 1995.

Danny Driess, Zhiao Huang, Yunzhu Li, Russ Tedrake, and Marc Toussaint. Learning multi-object dynamics with compositional neural radiance fields. In *Proc. of the Annual Conf. on Robot Learning (CoRL)*, volume 205 of *Proceedings of Machine Learning Research*, pages 1755–1768. PMLR, 14–18 Dec 2023.

Evan Drumwright and Dylan A. Shell. Modeling contact friction and joint friction in dynamic robotic simulation using the principle of maximum dissipation. In *Algorithmic Foundations of Robotics IX*, pages 249–266. Springer, Berlin, Heidelberg, 2010.

D. Drusvyatskiy, A. D. Ioffe, and A. S. Lewis. Curves of descent. *SIAM Journal on Control and Optimization*, 53(1):114–138, 2015. doi: 10.1137/130920216. URL https://doi.org/10.1137/130920216.

T. Erez, Y. Tassa, and E. Todorov. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4397–4404, 2015. doi: 10.1109/ICRA.2015.7139807.

Francisco Facchinei and Jong-Shi Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer-Verlag, New York, 2003.

Charbel Fares and Yskandar Hamam. Optimisation-based proximity queries and penetration depth computation. *Virtual Real.*, 13(2):131–136, jun 2009. ISSN 1359-4338.

N. Fazeli, Samuel Zapolsky, Evan Drumwright, and A. Rodríguez. Fundamental limitations in performance and interpretability of common planar rigid-body contact models. In *International Symposium on Robotics Research (ISRR)*, 2017a.

Nima Fazeli, Roman Kolbert, Russ Tedrake, and Alberto Rodriguez. Parameter and contact force estimation of planar rigid-bodies undergoing frictional contact. *The International Journal of*

Robotics Research, 36(13-14):1437–1454, 2017b. doi: 10.1177/0278364917698749.

Nima Fazeli, Samuel Zapolsky, Evan Drumwright, and Alberto Rodriguez. Learning data-efficient rigid-body contact models: Case study of planar impact. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 388–397. PMLR, 13–15 Nov 2017c. URL https://proceedings.mlr.press/v78/fazeli17a.html.

Nima Fazeli, Anurag Ajay, and Alberto Rodriguez. Long-horizon prediction and uncertainty propagation with residual point contact learners. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7898–7904. IEEE, 2020a.

Nima Fazeli, Samuel Zapolsky, Evan Drumwright, and Alberto Rodriguez. Fundamental limitations in performance and interpretability of common planar rigid-body contact models. In Nancy M. Amato, Greg Hager, Shawna Thomas, and Miguel Torres-Torriti, editors, *Robotics Research*, pages 555–571. Springer International Publishing, 2020b.

E.G. Gilbert, D.W. Johnson, and S.S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal on Robotics and Automation*, 4(2): 193–203, 1988. doi: 10.1109/56.2083.

Ch. Glocker and F. Pfeiffer. Multiple impacts with friction in rigid multibody systems. *Nonlinear Dynamics*, 7(4):471–497, jun 1995. ISSN 0924-090X.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. In Włodzisław Duch, Janusz Kacprzyk, Erkki Oja, and Sławomir Zadrożny, editors, *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, pages 799–804, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-28756-8.

Jayesh K. Gupta, Kunal Menda, Zachary Manchester, and Mykel Kochenderfer. Structured mechanical models for robot learning and control. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger, editors, *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 328–337. PMLR, 10–11 Jun 2020. URL https://proceedings.mlr.press/v120/gupta20a.html.

Mathew Halm and Michael Posa. Modeling and Analysis of Non-unique Behaviors in Multiple Frictional Impacts. In *Robotics: Science and Systems (RSS)*, Freiburg im Breisgau, Germany, 2019. URL http://roboticsproceedings.org/rss15/p22.pdf.

Mathew Halm and Michael Posa. Set-valued rigid body dynamics for simultaneous, inelastic, frictional impact. *Under Review*, 2023.

Xuchen Han, Joseph Masterjohn, and Alejandro Castro. A convex formulation of frictional contact

between rigid and deformable bodies, 2023.

Eric Heiden, David Millard, Erwin Coumans, Yizhou Sheng, and Gaurav S. Sukhatme. Neuralsim: Augmenting differentiable simulators with neural networks. *arXiv preprint arXiv:2011.04217*, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.

Peter C. Horak and Jeff C. Trinkle. On the similarities and differences among contact models in robot simulation. *IEEE Robotics and Automation Letters*, 4(2):493–499, apr 2019. ISSN 2377-3766.

Yildirim Hurmuzlu and Dan B. Marghitu. Rigid body collisions of planar kinematic chains with multiple contact points. *The International Journal of Robotics Research*, 13(1):82–92, feb 1994. ISSN 0278-3649.

Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4-5):698–721, 2021. doi: 10.1177/0278364920987859. URL https://doi.org/10.1177/0278364920987859.

A.P Ivanov. On multiple impact. *Journal of Applied Mathematics and Mechanics*, 59(6):887–902, jan 1995. ISSN 0021-8928.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, pages 12519–12530, 2019.

J. H. Jellet. *Treatise on the Theory of Friction*. Hodges, Foster and Co., Dublin, 1872.

Yifeng Jiang, Jiazheng Sun, and C. Karen Liu. Data-augmented contact model for rigid body simulation. *arXiv preprint arXiv:1803.04019*, 2018.

Wanxin Jin and Michael Posa. Task-driven hybrid model reduction for dexterous manipulation. *arXiv preprint arXiv:2211.16657*, November 2022.

Wanxin Jin, Alp Aydinoglu, Mathew Halm, and Michael Posa. Learning linear complementarity systems. In Roya Firoozi, Negar Mehr, Esen Yel, Rika Antonova, Jeannette Bohg, Mac Schwager, and Mykel Kochenderfer, editors, *Proceedings of The 4th Annual Learning for Dynamics and Control Conference (L4DC)*, volume 168 of *Proceedings of Machine Learning Research*, pages 1137–1149. PMLR, 23–24 Jun 2022. URL https://proceedings.mlr.press/v168/jin22a.html.

Aaron M Johnson, Samuel A Burden, and Daniel E Koditschek. A hybrid systems model for simple manipulation and self-manipulation systems. *The International Journal of Robotics Research*, 35

(11):1354–1392, sep 2016. ISSN 0278-3649.

Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

J. B. Keller. Impact With Friction. *Journal of Applied Mechanics*, 53(1):1–4, 03 1986. ISSN 0021-8936. doi: 10.1115/1.3171712.

Wisama Khalil and Etienne Dombre. *Modeling, Identification and Control of Robots*. Butterworth-Heinemann, Oxford, 2002. doi: https://doi.org/10.1016/B978-190399666-9/50006-3.

P. K. Khosla and T. Kanade. Parameter identification of robot dynamics. In *1985 24th IEEE Conference on Decision and Control*, pages 1754–1760, Dec 1985. doi: 10.1109/CDC.1985.268838.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations (ICLR)*, 2015.

S. Kolev and E. Todorov. Physically consistent state estimation and system identification for contacts. In *International Conference on Humanoid Robots (Humanoids)*, Nov 2015.

A. Ya. Kruger. On fréchet subdifferentials. *Journal of Mathematical Sciences*, 116(3):3325–3358, 2003. doi: 10.1023/A:1023673105317. URL https://doi.org/10.1023/A:1023673105317.

Charles E. Land. An evaluation of approximate confidence interval estimation methods for lognormal means. *Technometrics*, 14(1):145–158, 1972. ISSN 00401706.

Quentin Le Lidec, Igor Kalevatykh, Ivan Laptev, Cordelia Schmid, and Justin Carpentier. Differentiable simulation for physical system identification. *IEEE Robotics and Automation Letters*, February 2021. URL https://hal.archives-ouvertes.fr/hal-03025616.

Léon Lecornu. Sur la loi de coulomb. *Comptes Rendu des Séances de l'Academie des Sciences*, 140: 847–848, 1905.

R I Leine and N van de Wouw. *Stability and convergence of mechanical systems with unilateral constraints*. Springer Verlag, 2008.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018. URL http://jmlr.org/papers/v18/16-558.html.

Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *ICLR*, 2019.

Rong Liu, Hao Zhang, and James Busby. Convex hull covering of polygonal scenes for accurate collision detection in games. In *Proceedings of Graphics Interface 2008*, GI 2008, pages 203–210, Toronto, Ontario, Canada, 2008. Canadian Human-Computer Communications Society. ISBN 978-1-56881-423-0.

Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BklHpjCqKm.

Khaled Mamou. Volumetric hierarchical approximate convex decomposition. In Eric Lengyel, editor, *Game Engine Gems 3*, pages 141–158. A K Peters, 2016.

Khaled Mamou and Faouzi Ghorbel. A simple and efficient approach for 3d mesh approximate convex decomposition. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 3501–3504, 2009. doi: 10.1109/ICIP.2009.5414068.

Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. In *International Symposium on Robotics Research (ISRR)*, 2019.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. URL http://arxiv.org/abs/2003.08934v2.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2924–2932, Cambridge, MA, USA, 2014. MIT Press.

Jean-Jacques Moreau. Application of convex analysis to some problems of dry friction. In *Trends in Applications of Pure Mathematics to Mechanics, Vol. II (Second Symposium, Kozubnik, 1977)*, pages 263–280. Pitman, Boston, 1977.

Anusha Nagabandi, Kurt Konoglie, S. Levine, and V. Kumar. Deep dynamics models for learning dexterous manipulation. In *CoRL*, 2019.

Ngoc Son Nguyen and Bernard Brogliato. *Comparisons of Multiple-Impact Laws For Multibody Sys-*

tems: Moreau's Law, Binary Impacts, and the LZB Approach, pages 1–45. Springer International Publishing, Cham, 2018. doi: 10.1007/978-3-319-75972-2_1.

Armin Nurkanović, Sebastian Albrecht, Bernard Brogliato, and Moritz Diehl. The time-freezing reformulation for numerical optimal control of complementarity lagrangian systems with state jumps, 2021a.

Armin Nurkanović, Tommaso Sartor, Sebastian Albrecht, and Moritz Diehl. A time-freezing approach for numerical optimal control of nonsmooth differential equations with state jumps. *IEEE Control Systems Letters*, 5(2):439–444, 2021b. doi: 10.1109/LCSYS.2020.3003419.

Paul Painlevé. Sur les lois du frottment de glissment. *Comptes Rendu des Séances de l'Academie des Sciences*, 121:112–115, 1895.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Mihir Parmar*, Mathew Halm*, and Michael Posa. Fundamental Challenges in Deep Learning for Stiff Contact Dynamics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, March 2021. URL https://ieeexplore.ieee.org/document/9636383.

Bernd Pfrommer and Kostas Daniilidis. Tagslam: Robust slam with fiducial markers. *arXiv preprint arXiv:1910.00679*, 2019.

Samuel Pfrommer*, Mathew Halm*, and Michael Posa. ContactNets: Learning of Discontinuous Contact Dynamics with Smooth, Implicit Representations. In *The Conference on Robot Learning (CoRL)*, 2020. URL https://proceedings.mlr.press/v155/pfrommer21a.html.

T. Poggio and S. Smale. The mathematics of learning: Dealing with data. *Notices of the American Mathematical Society (AMS)*, 2003.

Michael Posa, Mark Tobenkin, and Russ Tedrake. Lyapunov Analysis of Rigid Body Systems with Impacts and Friction via Sums-of-Squares. In *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control (HSCC 2013)*, pages 63–72. ACM, apr 2013.

Michael Posa, Cecilia Cantu, and Russ Tedrake. A Direct Method for Trajectory Optimization of Rigid Bodies Through Contact. *International Journal of Robotics Research*, 33(1):69–81, jan 2014.

Michael Posa, Scott Kuindersma, and Russ Tedrake. Optimization and stabilization of trajectories for constrained dynamical systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2016-June, pages 1366–1373, Stockholm, Sweden, May 2016a. ISBN 9781467380263. doi: 10.1109/ICRA.2016.7487270. URL https://ieeexplore.ieee.org/abstract/

document/7487270/.

Michael Posa, Mark Tobenkin, and Russ Tedrake. Stability analysis and control of rigid-body systems with impacts and friction. *IEEE Transactions on Automatic Control (TAC)*, 61(6):1423–1437, 2016b.

C David Remy. Ambiguous collision outcomes and sliding with infinite friction in models of legged systems. *The International Journal of Robotics Research*, 36(12):1252–1267, oct 2017. ISSN 0278-3649.

Antônio H. Ribeiro, Koen Tiels, Jack Umenberger, Thomas B. Schön, and Luis A. Aguirre. On the smoothness of nonlinear system identification. *arXiv preprint arXiv:1905.00820*, 2019.

Antônio H. Ribeiro, Johannes N. Hendriks, Adrian G. Wills, and Thomas B. Schön. Beyond occam's razor in system identification: Double-descent when modeling dynamics. *arXiv preprint arXiv:2012.06341*, 2020.

E J Routh. *Dynamics of a System of Rigid Bodies*. MacMillan, London, 1891.

Caleb Rucker and Patrick M. Wensing. Smooth parameterization of rigid-body inertia. *IEEE Robotics and Automation Letters*, 7(2):2771–2778, 2022. doi: 10.1109/LRA.2022.3144517.

Walter Rudin. *Real and Complex Analysis*. McGraw-Hill Education, 1986.

Walter Rudin. *Functional Analysis*. McGraw-Hill Science/Engineering/Math, 1991.

Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4470–4479. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/sanchez-gonzalez18a.html.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Rolf Schneider. *Convex Bodies: The Brunn-Minkowski Theory (Encyclopedia of Mathematics and its Applications, Series Number 44)*. Cambridge University Press, Cambridge, England, February 1993.

Vlad Seghete and Todd D. Murphey. A propagative model of simultaneous impact: existence, uniqueness, and design consequences. *IEEE Transactions on Automation Science and Engineering*, 11(1):154–168, jan 2014. ISSN 1545-5955.

Breannan Smith, Danny M. Kaufman, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. Reflections on simultaneous impact. *ACM Transactions on Graphics*, 31(4):1–12, jul 2012. ISSN 07300301.

David Stewart and J. C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with Coulomb friction. *International Journal for Numerical Methods in Engineering*, 39(15):2673—2691, 1996.

David E. Stewart. Convergence of a time-stepping scheme for rigid body dynamics and resolution of painlevé's problems. *Archive Rational Mechanics and Analysis*, 145(3):215–260, 1998.

David E. Stewart. Rigid-body dynamics with friction and impact. *SIAM Review*, 42(1):3–39, 2000.

Yuval Tassa, Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, and Nicolas Heess. dm_control: Software and tasks for continuous control. *arXiv preprint arXiv:2006.12983*, 2020.

Ian H. Taylor, Siyuan Dong, and Alberto Rodriguez. Gelslim 3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger. In *ICRA*, 2022.

Russ Tedrake. Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation (course notes for mit 6.832). URL http://underactuated.mit.edu/.

Russ Tedrake. Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems, 2014. URL http://drake.mit.edu.

Emanuel Todorov. Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in MuJoCo. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 6054–6061, 2014. ISSN 10504729. doi: 10.1109/ICRA.2014.6907751.

Thomas K Uchida, Michael A Sherman, and Scott L Delp. Making a meaningful impact: modelling simultaneous frictional collisions in spatial multibody systems. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 471(2177):20140859–20140859, 2015. ISSN 1364-5021.

Gino Van den Bergen. Proximity queries and penetration depth computation on 3d game objects. In *Game developers conference*, volume 170, 2001.

Y.-T. Wang, V. Kumar, and J. Abel. Dynamics of rigid bodies undergoing multiple frictional contacts. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, volume 3, pages 2764–2769. IEEE Comput. Soc. Press, 1992.

Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics (TOG)*, 41(4):1–18, 2022.

Patrick M. Wensing, Michael Posa, Yue Hu, Adrien Escande, Nicolas Mansard, and Andrea Del Prete. Optimization-based control for dynamic legged robots. *arXiv preprint arXiv:2211.11644*, November 2022.

Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. TossingBot: Learning to throw arbitrary objects with residual physics. In *Robotics: Science and Systems (RSS)*, March 2019.

Guodong Zhang, Chaoqi Wang, Bowen Xu, and Roger Grosse. Three mechanisms of weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1lz-3Rct7.